

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«На правах рукопису»
УДК 004.582

«До захисту допущено»

Завідувач кафедри

_____ І.А. Дичка

«__» _____ 2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення

**на тему: «Спосіб та програмне забезпечення прогнозування
користувацьких дій при конструюванні схем бізнес-процесів
виробництва»**

Виконав:

студент VI курсу, групи КП-71мп

Шевчук Євгеній Олегович _____

Керівник:

Старший викладач кафедри ПЗКС, к.т.н.,

Люшенко Л.А. _____

Рецензент:

Доцент кафедри ММСА, к.т.-м.н., доцент

Шубенкова І.А. _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2018 року

**Національний технічний університет України
“Київський політехнічний інститут
імені Ігоря Сікорського”**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Напрямок підготовки 6.050103 “Програмна інженерія”

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І.А. Дичка

“___” _____ 201__ р.

**З А В Д А Н Н Я
на магістерську дисертацію**

студенту групи КП-71мп Шевчуку Євгенію Олеговичу

1.Тема дисертації «СПОСІБ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ СХЕМ БІЗНЕС-ПРОЦЕСІВ ВИРОБНИЦТВА» затверджена наказом по університету № _____ від «___» _____ 201__ р.

2.Термін подання студентом проекту: «___» _____ 201__ р.

3.Об’єкт дослідження: процес автоматизації прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва

4.Предмет дослідження: спосіб та програмне забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів

5.Перелік задач, які мають бути вирішені:

- оцінити сучасний стан проблеми, обґрунтувати актуальність напрямку досліджень, сформулювати мету та задачі дослідження;
- описати сферу розповсюдження машинного навчання, навести загальну ідею опису прецедентів інформаційної одиниці, проаналізувати типові задачі, область використання та структури для зберігання прецедентного опису;

- сформулювати вимоги до способу та програмного забезпечення прогнозування користувацьких дій під час конструювання схем бізнес-процесів, описати їх особливості та детально описати спосіб прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва;
- проаналізувати особливості програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів, описати принципи роботи запропонованого способу та деталі реалізації його алгоритму;
- виконати тестування розробленого програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів та проаналізувати отримані результати;
- описати бізнес модель, що дозволить представити на ринку повноцінний програмний продукт із використанням представлених у роботі напрацювань.

6.Перелік обов’язкового ілюстративного матеріалу:

- блок-схема алгоритму побудови дерева прецедентів;
- блок-схема алгоритму побудови списку вірогідних до виконання користувачем дій, що будується залежно від дерева прецедентів;
- блок-схема алгоритму ідентифікації виконаної користувачем дії;
- блок-схема аналізуючого стан схеми бізнес-процесів алгоритму.

7.Рецензент – к.ф-м.н., доцент Шубенкова І.А.

8.Дата видачі завдання: «___» _____ 201__ р.

Науковий керівник _____ **Л.А. Люшенко**

Завдання прийняв до виконання _____ **Є.О. Шевчук**

РЕФЕРАТ

Актуальність теми. В створенні програмного забезпечення для управління різними видами господарської діяльності підприємства дуже велику частину витрат робочого часу займає моделювання бізнес-процесів. Зазвичай розробники нехтують стандартизованим процесом збору вимог до автоматизованої системи та детальною побудовою моделі системи. Нехтування цими етапами створення програмного забезпечення призводить до проблем в проектуванні та розробці системи, а також унеможлиблює успішну реалізацію проекту автоматизації загалом. В зв'язку з цим актуальним є розв'язання проблеми пришвидшення конструювання релевантних бізнес моделей програмного забезпечення за рахунок автоматизації процесу прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.

Об'єктом дослідження є процес автоматизації прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.

Предметом дослідження є спосіб та програмне забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів.

Мета роботи полягає у розробці способу та програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.

Методи дослідження. В даній роботі використовуються методи теоретичного дослідження: аналіз, синтез та узагальнення. Також застосовувалися емпіричні методи: експеримент, спостереження, вимірювання та опис.

Наукова новизна полягає в розробці нового способу виконання користувацьких дій конструювання схем бізнес-процесів виробництва за рахунок автоматизації процесу прогнозування цих дій. Цей спосіб, на відміну від існуючих, дозволяє прискорити конструювання схем бізнес-процес не менше ніж на 20%.

Практична цінність отриманих результатів роботи полягає в тому, що запропоновані спосіб та програмне забезпечення прогнозування користувацьких дій може використовуватися при створенні програмного забезпечення з метою підвищення ефективності процесу розробки програмного забезпечення.

Апробація роботи. Основні положення та результати роботи були представлені та обговорювались на XI науковій конференції магістрантів та аспірантів “Прикладна математика та комп’ютинг” ПМК-2018.

Структура та обсяг роботи. Магістерська дисертація складається зі вступу, п’яти розділів, висновків та додатків.

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі дослідження.

У першому розділі описано сферу розповсюдження машинного навчання, наведено загальну ідею опису прецедентів інформаційної одиниці, проаналізовано типові задачі, область використання та структури для зберігання прецедентного опису.

У другому розділі сформовано вимоги до способу та програмного забезпечення прогнозування користувацьких дій під час конструювання схем бізнес-процесів, описано їх особливості та детально описано спосіб прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.

У третьому розділі представлено особливості програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів, більш детально описано принципи роботи запропонованого способу та деталі реалізації його алгоритму.

У четвертому розділі представлено результати тестування розробленого програмного забезпечення прогнозування користувацьких дій під час конструювання схем бізнес-процесів, представлено аналіз отриманих результатів та наведено шляхи до вдосконалення програмного забезпечення.

У п'ятому розділі запропоновано бізнес модель, яка дозволяє представити на ринку повноцінний програмний продукт із використанням представлених у роботі напрацювань.

У висновку проаналізовано отримані результати даної магістерської роботи.

У додатках наведено дерево проблем, порівняння потенційних користувачів та канву згідно запропонованої бізнес-моделі.

Ключові слова: прогнозування, програмне забезпечення, дії користувача, схеми бізнес-процесів, виробництво.

ABSTRACT

Actuality of theme. In the creation of software for managing various types of economic activities of the enterprise a very large part of the cost of working time is the simulation of business processes. Usually, developers ignore the standardized process of collecting requirements for an automated system and detailed construction of the system model. Neglecting these stages of software creation leads to problems in designing and developing the system, and also prevents the successful implementation of the automation project in general. In this regard, it is important to solve the problem of speeding up the design of relevant business software models by automating the process of predicting user actions when designing schemes for business processes of production.

Object of research is the process of automation of prediction of user actions in the design of schemes business processes of production.

Subject of research is a method and software for predicting user actions in the design of business process diagrams.

Research objective is to develop a method and software for predicting user actions in the design of business process diagrams of production.

Research methods. In this paper, the methods of theoretical research are used: analysis, synthesis and generalization. Also used empirical methods: experiment, observation, measurement and description.

Scientific novelty consists in developing a new way of performing custom actions for designing schemes of business processes of production by automating the process of forecasting these actions. This method, in contrast to existing ones, allows accelerating the design of business process diagrams by at least 20%.

The practical value of the results obtained is that the proposed method and software for predicting user actions can be used to create software to enhance the effectiveness of the software development process.

Approbation. The main provisions and results of the work were presented and discussed at the 11th scientific conference of graduate students and postgraduates "Applied Mathematics and Computer" PMK-2018.

Structure and content of the thesis. The master's thesis consists of an introduction, five sections, conclusions and appendices.

The introduction provides a general description of the work, evaluated the current state of the problem, substantiated the relevance of the research direction, formulated the purpose and objectives of the study.

The first chapter describes the scope of machine learning, describes the general idea of describing the precedents of the information unit, analyzes the typical tasks, area of use and structures for storing a case description.

In the second section, requirements for the method and software for predicting user actions are generated during the design of business process diagrams, describes their features and describes in detail the way to predict user actions in designing business process diagrams of production.

The third section presents features of software for predicting user actions in the design of business process diagrams, describes in more detail the principles of the proposed method and the details of the implementation of its algorithm.

The fourth section presents the results of the testing of the developed software for predicting user actions in the design of business process diagrams, analyzes the results obtained and provides ways to improve the software.

The fifth section proposes a business model that allows you to submit a complete software product to the market using the work presented in the workflow.

The conclusion is analyzed the results of this master's work.

The applications have a problem tree, a comparison of potential users and a canvas according to the proposed business model.

Keywords: forecasting, software, user actions, schemes of business processes, production.

РЕФЕРАТ

Актуальность темы. В создании программного обеспечения для управления различными видами хозяйственной деятельности предприятия очень большую часть затрат рабочего времени занимает моделирование бизнес-процессов. Обычно разработчики пренебрегают стандартизированным процессом сбора требований к автоматизированной системе и детальным построением модели системы. Пренебрежение этими этапами создания программного обеспечения приводит к проблемам в проектировании и разработке системы, а также исключает успешную реализации проекта автоматизации в целом. В связи с этим актуальным является решение проблемы ускорения конструирования релевантных бизнес моделей программного обеспечения за счет автоматизации процесса прогнозирования пользовательских действий при конструировании схем бизнес-процессов производства.

Объектом исследования является процесс автоматизации прогнозирования пользовательских действий при конструировании схем бизнес-процессов производства.

Предметом исследования является способ и программное обеспечение прогнозирования пользовательских действий при конструировании схем бизнес-процессов.

Цель работы заключается в разработке способа и программного обеспечения прогнозирования пользовательских действий при конструировании схем бизнес-процессов производства.

Методы исследования. В данной работе используются методы теоретического исследования: анализ, синтез и обобщение. Также применялись эмпирические методы: эксперимент, наблюдение, измерение и описание.

Научная новизна заключается в разработке нового способа выполнения пользовательских действий конструирования схем бизнес-процессов производства за счет автоматизации процесса прогнозирования этих действий. Этот способ, в отличие от существующих, позволяет ускорить конструирование схем бизнес-процессов не менее чем на 20%.

Практическая ценность полученных результатов работы заключается в том, что предложенные способ и программное обеспечение прогнозирования пользовательских действий может использоваться при создании программного обеспечения с целью повышения эффективности процесса разработки программного обеспечения.

Апробация работы. Основные положения и результаты работы были представлены и обсуждались на XI научной конференции магистрантов и аспирантов "Прикладная математика и компьютеринг" ПМК-2018.

Структура и объем работы. Магистерская диссертация состоит из введения, пяти глав, заключения и приложений.

Во введении дана общая характеристика работы, выполнена оценка современного состояния проблемы, обоснована актуальность направления исследований, сформулированы цели и задачи исследования.

В первом разделе описано сферу распространения машинного обучения, приведена общая идея описания прецедентов информационной единицы, проанализированы типовые задачи, область использования и структуры для хранения прецедентного описания.

Во втором разделе сформированы требования к способу и программному обеспечению прогнозирования пользовательских действий при конструировании схем бизнес-процессов, описаны их особенности и подробно описан способ прогнозирования пользовательских действий при конструировании схем бизнес-процессов производства.

В третьем разделе представлены особенности программного обеспечения прогнозирования пользовательских действий при конструировании схем бизнес-процессов, более подробно описаны принципы работы предлагаемого способа и детали реализации его алгоритма.

В четвертом разделе представлены результаты тестирования разработанного программного обеспечения прогнозирования пользовательских действий при конструировании схем бизнес-процессов, представлен анализ полученных результатов и приведены пути к совершенствованию программного обеспечения.

В пятом разделе предложено бизнес модель, которая позволяет представить на рынке полноценный программный продукт с использованием представленных в работе наработок.

В заключении проанализированы полученные результаты данной магистерской работы.

В приложениях приведены дерево проблем, сравнение потенциальных пользователей и канву согласно предложенной бизнес-модели.

Ключевые слова: прогнозирование, программное обеспечение, действия пользователя, схемы бизнес-процессов, производство.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І.Сікорського”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

ПОЯСНЮВАЛЬНА ЗАПИСКА

зі спеціальності 8.04030101 “Прикладна математика”

на тему: СПОСІБ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ
КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНТРУЮВАННІ СХЕМ
БІЗНЕС-ПРОЦЕСІВ ВИРОБНИЦТВА

Студент КП-71мп Шевчук Євгеній Олегович _____

Науковий керівник старший викладач, к.т.н., Люшенко Л.А. _____

ЗМІСТ

ВСТУП	4
1. АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ РОБОТІ З ВЕБ-ДОДАТКАМИ	6
1.1. Сфера розповсюдження машинного навчання	7
1.2. Загальна ідея опису прецедентів інформаційної одиниці.....	8
1.3. Типові задачі та область використання прецедентного опису	11
1.4. Структури зберігання результатів прецедентного опису	12
1.5. Висновки	15
2. СПОСІБ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ СХЕМ БІЗНЕС-ПРОЦЕСІВ.....	16
2.1. Вимоги до способу та програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.....	16
2.2. Особливості прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва	18
2.3. Спосіб прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва	20
2.4. Висновки	28
3. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ АЛГОРИТМІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ	30
3.1. Архітектура програмного забезпечення	30
3.2. Опис інтерфейсу веб-додатку	32
3.3. Засоби та технології реалізації програмного забезпечення.....	34
3.4. Вимоги до платформи виконання програмного забезпечення.....	35
3.5. Особливості реалізації головних алгоритмів та структур даних	36
3.6. Висновки	45

4. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ	47
4.1. Тестування розробленого програмного забезпечення	47
4.2. Аналіз розробленого програмного застосунку	48
4.3. Вдосконалення розробленого програмного забезпечення.....	51
4.4. Висновки	54
5. ПОБУДОВА БІЗНЕС-МОДЕЛІ	56
5.1. Опис проблеми	56
5.2. Зацікавлені сторони	57
5.3. Опис рішення проблеми	58
5.4. Бізнес-продукт. Основні характеристики бізнес-продукту	58
5.5. Конкурентні переваги продукту	60
5.6. Клієнти. Сегменти ринку.....	61
5.7. Унікальна ціннісна пропозиція.....	62
5.8. Доходи і витрати	62
5.9. Висновки	64
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	69
ДОДАТКИ.....	71

ВСТУП

В сучасних умовах досить важко уявити підприємство без автоматизованих систем управління його господарською діяльністю. Створення програмного забезпечення для управління різними видами господарської діяльності підприємства, починаючи з маркетингу та продажу і закінчуючи виробництвом та логістикою, є дуже витратним проектом з точки зору грошових та часових витрат. При цьому етап збору вимог та побудови моделі автоматизованої системи управління діяльністю підприємства може зайняти 25% від усього часу на розробку та впровадження такої системи.

Зазвичай розробники нехтують стандартизованим процесом збору вимог до автоматизованої системи та детальною побудовою моделі системи. Проте, збір вимог та побудова моделі є тими етапами, які гарантують якість системи та задоволеність зацікавлених сторін у впровадженні такої системи. Нехтування цими етапами розробки призводить до проблем конструювання та розробки системи, а також унеможлиблює успішну реалізацію проекту автоматизації загалом.

Відповідно до цього, метою дослідження є спрощення процесу моделювання бізнес-процесів для систем автоматизації управління діяльністю підприємства за рахунок автоматизації технології прогнозування користувацьких дій при моделюванні бізнес-процесів та скорочення термінів реалізації цього процесу, а саме:

- зменшення кількості користувацьких помилок при роботі з конструктором схем бізнес-процесів не менше, ніж на 40%;
- точність роботи прогнозуючого модуля повинна бути не менше, ніж 85%;

- скорочення загальних термінів реалізації моделювання не менше, ніж на 25%.

Таким чином, основною задачею даної магістерської роботи є створення способу прогнозування користувацьких дій при побудові схем бізнес-процесів виробництва та побудова відповідного програмно-аналітичного модуля.

1 АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ РОБОТІ З ВЕБ-ДОДАТКАМИ

Автоматизація людської діяльності при взаємодії з програмними застосунками є доволі складною галуззю, що включає в себе велику кількість часткових випадків, що мають унікальні вимоги до поведінки програмного застосунку. Саме тому ця сфера не може бути описана набором правил та умов, тобто звичайним структурним алгоритмом.

Для вирішення поставленої задачі необхідно використовувати більш гнучкі методи, що дозволять прогнозуючому програмному застосунку розширювати власний набір «знань» під час взаємодії із користувачем.

Так як основною метою автоматизації є саме побудова схем бізнес-процесів виробництва, то зазначена вище умова, виставлена до способу прогнозування користувацьких дій, також повинна бути доповнена наступними [1]:

- прогнозування користувацьких дій повинно базуватися на виконаних користувачем діях;
- дія, що може бути допущена для аналізу прогнозуючим програмним застосунком, повинна мати чіткий безпомилковий кінцевий результат в рамках моделювання схем бізнес-процесів виробництва;
- набір прогнозованих дій повинен вираховуватись на підставі набору попередніх користувацьких дій;
- результат виконання прогнозованої дії не може виходити за рамки результатів дій, виконаних користувачем до часу прогнозування;
- результатом виконання прогнозованої дії може біти тільки чинна схема бізнес-процесів;

Як відомо, кожен проект з моделювання схем бізнес-процесів виробництва є унікальним, та має власні тонкості та обмеження. Саме тому зазначені умови є достатньо обмежуючими для прогнозуючого програмного застосунку в плані надання вільного формату кінцевого результату прогнозованих дій.

Зазначені умови є необхідною мірою для гарантування чіткості та відсутності помилок в результатах прогнозованих дій, які може застосувати користувач в рамках свого унікального проекту.

Оскільки рішення поставленої задачі повинне мати гнучкий алгоритм прогнозування, що може розширюватися та навчатися на отриманих від користувача даних, та, водночас, мати чітко зазначені межі роботи, то найбільш вдалим вибором для реалізації рішення є використання штучного інтелекту, а саме, його часткового випадку – алгоритмів машинного навчання.

1.1 Сфера розповсюдження машинного навчання

Машинне навчання – клас методів штучного інтелекту, характерною рисою яких є не пряме вирішення задачі, а навчання в процесі застосування рішень багатьох подібних задач.

Розрізняють два основних типи навчання [2]:

- індуктивне навчання (прецедентний опис);
- дедуктивне навчання.

Проаналізувавши зазначені типи навчання, можна зазначити, що прецедентний опис використовується у випадках, коли необхідно знайти закономірності в наборі даних та побудувати подальший алгоритм обробки початкових даних, а дедуктивне навчання використовується у випадках

формалізації експертних даних та переносі їх у електронний вигляд для подальшого використання або вдосконалення.

Тож, виходячи із наведених описів, можна зробити висновок, що для побудови алгоритму прогнозування користувацьких дій слід використовувати обробку користувацьких дій за допомогою машинного навчання з використанням прецедентного опису об'єктів.

1.2 Загальна ідея опису прецедентів інформаційної одиниці

Опис прецедентів інформаційної одиниці (або прецедентний опис) – це n -вимірний вектор ознак, що представляється, частіше за все, числовими даними та використовується в розділі машинного навчання. Для опису ознак, частіше за все, обирається саме числове представлення кожної з властивостей об'єктів інформації адже такий формат полегшує подальші обробку та статистичний аналіз даних [3].

Наприклад, при представленні у вигляді опису прецедентів аудіо-файлів у ролі числових ознак можуть виступати кожний з часових проміжків, що досліджується, також до нього може бути додано значення частоти аудіо-сигналі, гучність доріжки в даний часовий проміжок та інші.

При подальшому дослідженні інформації кожні з ознак розширюються коефіцієнтами, що додаються до загального опису ознак та використовуються при аналізі та структуруванні ознак під час формування результатів машинного навчання. Прецедентний опис ознак вважається потужною технологією, порівняно з іншими методами аналізу, які використовувалися раніше, чим дозволяє значно скоротити аналіз інформації в задачах високих розмірностей.

Загалом процес опису прецедентів інформаційної одиниці можна розділити на три підпроцеси:

- виділення ознак;
- вибір ознак;
- зниженні розмірності ознак.

Розглянемо більш детально кожний з підпроцесів.

1.2.1 Виділення ознак

Виділення ознак – процес перетворення вхідних даних алгоритму на скорочений набір ознак у випадку, коли ці дані признаються надлишковими та не можуть використовуватися для ефективного аналізу. В свою чергу, вхідними даними для процесу виділення ознак є набір даних вимірювань, з якого в подальшому будуються похідні значення, які мають бути ненадлишковими та інформативними в контексті роботи алгоритму [4].

Успішно виділеними ознаками вважаються ті, які мають достатню кількість необхідної інформації для виконання над ними, замість початкових надлишкових даних, заданої роботи із позитивним кінцевим результатом. Основною проблемою, яка виникає при виконанні процесу виділення ознак, є надмірне споживання алгоритмом машинної пам'яті та обчислювальних потужностей. Для вирішення зазначеної проблеми існують спеціальні алгоритми, що пришвидшують виконання роботи та економлять використання ресурсів.

До найбільш популярних алгоритмів можна віднести наступні [4]:

- метод головних компонент;
- приховано-семантичний аналіз.

Наведемо скорочений опис кожного з алгоритмів.

Метод головних компонент – алгоритм, що використовує ортогональне перетворення множини спостережень з можливо пов'язаними змінними, та базується на логічному припущенні, що значення множини взаємозалежних ознак породжують деякий загальний результат [4].

Приховано-семантичний аналіз – метод обробки інформації, що дозволяє аналізувати взаємозв'язок між набором властивостей, що зустрічаються в інформаційних одиницях, шляхом створення семантичного «словника» похідних властивостей, який в результаті алгоритму трансформується у набір ознак [5].

1.2.2 Вибір ознак

Вибір ознак – процес фільтрування вхідних ознак від недоречних, тобто тих, що є надлишковими та неінформативними, та в розрізі конкретного дослідження мають низьку корисність [6].

Наведемо основні методики, що використовуються в процесі вибирання ознак:

- надання набору спрощеного вигляду для поліпшення інтерпретування алгоритмом або науковцем;
- поліпшення узагальнення ознак через зменшення кількості кроків обробки.

Головною умовою використання процесу вибору ознак є наявність великої кількості недоречних вхідних даних. В такому випадку за результатами роботи цього процесу вибирання недоречні дані можуть бути усунуті без впливу на кінцевий результат роботи групи процесів прецедентного опису.

Слід зазначити, що процеси виділення та вибирання можуть здатися подібними між собою, але це не так. Результатом процесу виділення ознак є модифіковані за певним правилом початкові набори даних, а результатом,

відповідно, вибирання ознак, є підмножина початкових ознак, що не піддається модифікуванню.

Власне, основним недоліком цього підходу є відсутність узагальнених алгоритмів, що використовуються для процесу вибору ознак, адже цей процес є дуже індивідуальним та формується під чітко поставлену задачу, тому єдиного рішення для виконання процесу вибирання ознак не існує.

1.2.3 Зниження розмірності ознак

Зниження розмірності опису ознак – процес, що комбінує в собі роботу процесів виділення та вибирання, результатом якого є об'єднані результати зазначених процесів в скороченому вигляді [7]. Під скороченим виглядом мається на увазі застосування до результату попередні процесів алгоритмів стискання великого набору даних, результатом яких є модифікований вхідний ознаковий опис інформаційної структури та словник, який використовується для дешифрування модифікованого набору даних для більш чіткого розуміння дослідником.

1.3 Типові задачі та область використання прецедентного опису

За допомогою прецедентного опису, що є складовою частиною алгоритмів машинного навчання, на теперішній час вирішується велика кількість задач в технічних областях діяльності. Основними задачами, притаманними технічній області, є :

- розпізнавання об'єктів за допомогою комп'ютерного зору;
- розпізнавання людської мови;
- розпізнавання печатного та непечатного тексту;
- фільтрування електронної пошти;
- категоризація документів.

Зазначені задачі належать до різних технічних областей, та мають принципово різні за смисловим навантаженням початкові дані та результати, але всі вони вирішуються за допомогою використання прецедентного опису до початкових даних задачі за методом вирішення задач «навчання з вчителем».

Проаналізувавши зазначений тип задач, можна відзначити, що суть її полягає в тому, що кожний прецедент представляє собою пару «об'єкт – відповідь», а завдання, яке ставиться до задачі – знайти функціональну залежність відповідей від описів об'єктів і побудувати алгоритм. Таким чином, побудований алгоритм повинен приймати на вході об'єкт з даними, а в результаті повертати побудовану модель, залежно від початкових даних [8].

1.4 Структури зберігання результатів прецедентного опису

Слід зауважити, що кожна з існуючих структур даних має власне, найбільш корисне з точки зору економії пам'яті, простоти та швидкодії, використання в задачі конкретного типу, але в іншій задачі може лише заважати в реалізації алгоритму, через що розв'язання задачі вийде надто складне та повільне.

Проаналізуємо найбільш популярні структури зберігання даних при реалізації опису ознак.

До найбільш популярних структур даних можна віднести наступні:

- масив;
- зв'язний список;
- дерево;
- хеш-таблиця.

1.4.1 Масив

Масив – упорядкований набір даних, що використовується для зберігання елементів різних типів даних, що ідентифікується за допомогою одного чи більше індексів. Також масиви можуть відрізнятися порядком входження та взяття елементів: стек та черга. Стек реалізує правило «перший зайшов – останній пішов», черга реалізує правило «перший зайшов, першим пішов» [10].

Так як масив – це найпростіша структура даних, що при великій кількості елементів стає складною та повільною в обробці, використовується вона в простих задачах будь-якого типу з невеликою кількістю початкових та кінцевих даних.

1.4.2 Зв’язний список

Зв’язний список – динамічна структура даних, що складається з окремих вузлів, кожен з яких зберігає набір даних та два посилання на попередній та/або наступний елементи списку [11].

До головних переваг зв’язних списків можна віднести структурну гнучкість, невеликі затрати часу на виконання операцій додавання та видалення елементів та здатність зберігати необмежений об’єм інформації. Серед недоліків зв’язних списків слід зазначити складність прямого доступу до елементів, важкість організації паралельного використання списку та кешування елементів.

З виявлених переваг та недоліків можна зробити висновок, що зв’язні списки прекрасно підходять під задачі з великими об’ємами початкових та кінцевих даних та невеликою кількістю елементів, що редагуються під час виконання.

1.4.3 Дерево

Дерево – це ієрархічна структура даних, емуляюча деревовидну структуру, що складається з вузлів та ребер, що їх поєднують. По своїй суті, дерева являються зв'язним графами, що розповсюджуються в одному напрямку та мають неорієнтовані ребра однакової ваги [12].

Проаналізувавши основні властивості дерев, можна заявити, що ця структура є найбільш зручною для організації кінцевих даних при роботі з прецедентним описом через її гнучку структуру та швидкість редагування даних.

1.4.4 Хеш-таблиця

Хеш-таблиця – це структура даних, що дозволяє зберігати пари «ключ, значення» і виконувати над даними, що зберігаються три основні операції: додавання, видалення та пошук за ключем [13].

Загалом, хеш-таблиці мають високі показники швидкодії при виконанні з ними операцій додавання, видалення та пошуку елементів. Тому, зважаючи на принцип зберігання даних, хеш-таблиці можуть використовуватися в задачах з великою кількістю структурованих початкових даних.

Проаналізувавши основні переваги та недоліки зазначених структур даних, можна чітко виділити ті, що найбільш підходять для поставленої задачі прогнозування користувацьких дій, а саме: масив та дерево. Зазначені структури обрані через їх простоту в реалізації, ефективність та швидкість в обробці.

1.5 Висновки

Виходячи з загальної ідеї прецедентного опису об'єктів задача прогнозування користувацьких дій при конструюванні схем бізнес-процесів зводиться до динамічного формування моделі поведінки конкретного користувача та подальшого аналізу побудованої моделі.

Для побудови моделі найбільш підходящим методом можна відзначити метод, притаманний задачі «навчання з вчителем» через його динамічність та швидкодію.

Прецедентний опис ознак вважається потужною технологією, порівняно з іншими методами аналізу, які використовувалися раніше, чим дозволяє значно скоротити аналіз інформації в задачах високих розмірностей.

Структурами зберігання даних для алгоритму, що буде побудовано, слід обрати масив даних та дерево, через їх простоту реалізації, ефективність та швидкість в обробці.

2 СПОСІБ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ СХЕМ БІЗНЕС-ПРОЦЕСІВ

2.1 Вимоги до способу та програмного забезпечення прогнозування користувачьких дій при конструюванні схем бізнес-процесів виробництва

Для збору користувачьких вимог було проведено аналіз існуючих методів та програмних забезпечень прогнозування користувачьких дій. Також було проведено онлайн анкетування серед незалежних потенційних користувачів програмного забезпечення прогнозування користувачьких дій при моделюванні схем бізнес-процесів виробництва, що включало в себе питання стосовно корисної функціональності програмного забезпечення з урахуванням пріоритетів кожної з функціональних вимог.

Програмне забезпечення повинне:

- мати графічний інтерфейс для взаємодії зі списком згенерованих дій;
- пропонувати користувачеві виконати в автоматичному режимі найбільш вірогідні дії, що залежать від попередньо виконаних;
- надавати можливості аналізу побудованої схеми бізнес-процесів на наявність помилок;
- надавати можливість повернення схеми бізнес-процесів до попереднього стану після виконання прогнозованої дії;
- будувати список прогнозованих користувачьких дій не довше однієї секунди після завершення виконання попередньої дії;
- будувати список прогнозованих користувачьких дій основуючись на послідовності виконаних користувачьких дій;
- використовувати для прогнозування користувачьких дій оптимізовані, в плані швидкодії та використання комп'ютерної пам'яті, структури даних.

Наведений список містить в собі вимоги до способу прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва, а саме:

- побудова списку прогнозованих дій повинна відбуватися на основі послідовності виконаних користувацьких дій;
- побудова списку прогнозованих дій повинна займати не довше однієї секунди після завершення виконання користувачем попередньої дії;
- для побудови списку прогнозованих дій необхідно використовувати оптимізовані за використанням машинних ресурсів структури даних.

Прогнозуючий модуль після виконання будь-якої дії моделювання схеми бізнес-процесів повинен повертати користувачеві прогнозований список подальших дій. Прогнозування вірогідних дій повинно вираховуватися на основі дерева прецедентів, що будується під час всього часу роботи із програмним забезпеченням на основі описів виконаних користувацьких дій, та послідовності декількох останніх дій. Побудова списку вірогідних користувацьких дій повинна займати не більше однієї секунди процесорного часу та бути оптимізованою в плані використання пам'яті робочої машини користувача. Також, перед додаванням інформації стосовно виконаної користувачем дії, актуальна схема бізнес-процесів виробництва повинна проходити перевірку на наявність помилок в структурі схеми.

Стосовно вимог до графічної частини слід зазначити, що програмне забезпечення повинно мати графічний інтерфейс для взаємодії із списком прогнозованих користувацьких дій та мати можливість повернення схеми бізнес-процесів виробництва до попереднього стану в будь-який момент роботи із додатком.

2.2 Особливості прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва

Основною задачею способу прогнозування є виконання обчислень у вигляді побудови дерева прецедентів та виконання пошуку входження ланцюга користувацьких дій в дерево. Зазначені обчислення є доволі складними, тому можуть займати відносно великий, до 1000 мілісекунд, проміжок часу роботи процесора. Оскільки запуск програмного забезпечення прогнозування користувацьких дій виконується на однопоточній платформі веб, під час виконання обчислень графічна частина програмного забезпечення прогнозування користувацьких дій буде блокувати свою роботу та не зможе взаємодіяти із користувачем.

Виходячи з цього, спосіб прогнозування повинен забезпечувати необхідність виконання обчислень без блокування основного потоку виконання програмного забезпечення.

Досягнення вищезазначеною вимоги можливе двома способами:

- використання переривань обчислень при прогнозування користувацьких дій;
- використання додаткового браузерного потоку обчислень.

Проаналізувавши зазначені способи, можна зробити висновок, що використання переривань обчислень в програмному застосунку є більше технічним «трюком», ніж технологією паралельного виконання програмного забезпечення, адже його реалізація зводиться к асинхронному запуску окремих складових частин алгоритму та виконання під час утвореної асинхронної паузи дій, що накопилися з графічної частини програмного застосунку. Таке рішення повністю не вирішує проблеми блокування користувацького інтерфейсу, саме тому було обрано використання додаткового браузерного потоку виконання за допомогою технології вебворкерів.

Суть даного методу полягає в передачі управління над виконанням обчислень додатковому браузерному потоку та асинхронне очікування результатів обчислень в основному потоці. Таким чином, основний потік веб-додатку не блокується та може обробляти дії користувача і в час, коли вебворкер завершить роботу над обчисленням, отримати результати та відобразити їх користувачеві.

Схема взаємодії основного потоку із додатковим зазначена на рис. 1.



Рис. 1 – Схема взаємодії основного та додаткового потоків виконання

2.3 Спосіб прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва

Процес прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва можливо розбити на з три модулі: аналізуючий, ідентифікуючий та прогнозуючий.

2.3.1 Аналізуючий модуль

Аналізуючий модуль призначений для послідовного аналізу схеми бізнес-процесів виробництва, що конструюється користувачем під час роботи із додатком, на наявність помилок.

За допомогою даної модуля виявляються допущені помилки наступних типів:

- відсутність початкових даних/результатів роботи;
- відсутність виконавців та ресурсів роботи;
- наявність незатунельованих стрілок на схемі;
- наявність незастосованих до взаємодії елементів на схемі;
- відсутність підписів елементів на схемі.

З точки зору реалізації, аналізуючий модуль буде послідовним алгоритмом, який буде виконуватися по факту виконання користувачем будь-якої зареєстрованої дії.

Вхідними даними для алгоритму виступатиме опис схеми бізнес-процесів виробництва в форматі багаторівневого об'єкту, відсортованого в порядку вкладеності рівнів та кожного з елементів конкретного рівня, що будується за допомогою вже реалізованої функціональності в конструкторі схем бізнес-процесів в рамках виконання бакалаврського диплому.

Робота алгоритму полягатиме в рекурсивному переборі отриманого об'єкту, перевірці кожного з рівнів схеми бізнес-процесів на відповідність

заданим правилам та поверненням результату перевірки, від якого надалі залежить, чи буде дію, що спровокувала зміну стану схеми бізнес-процесів, використано в прогнозуючому модулі веб-додатку.

Блок-схему аналізуючого алгоритму наведено на рис. 2.

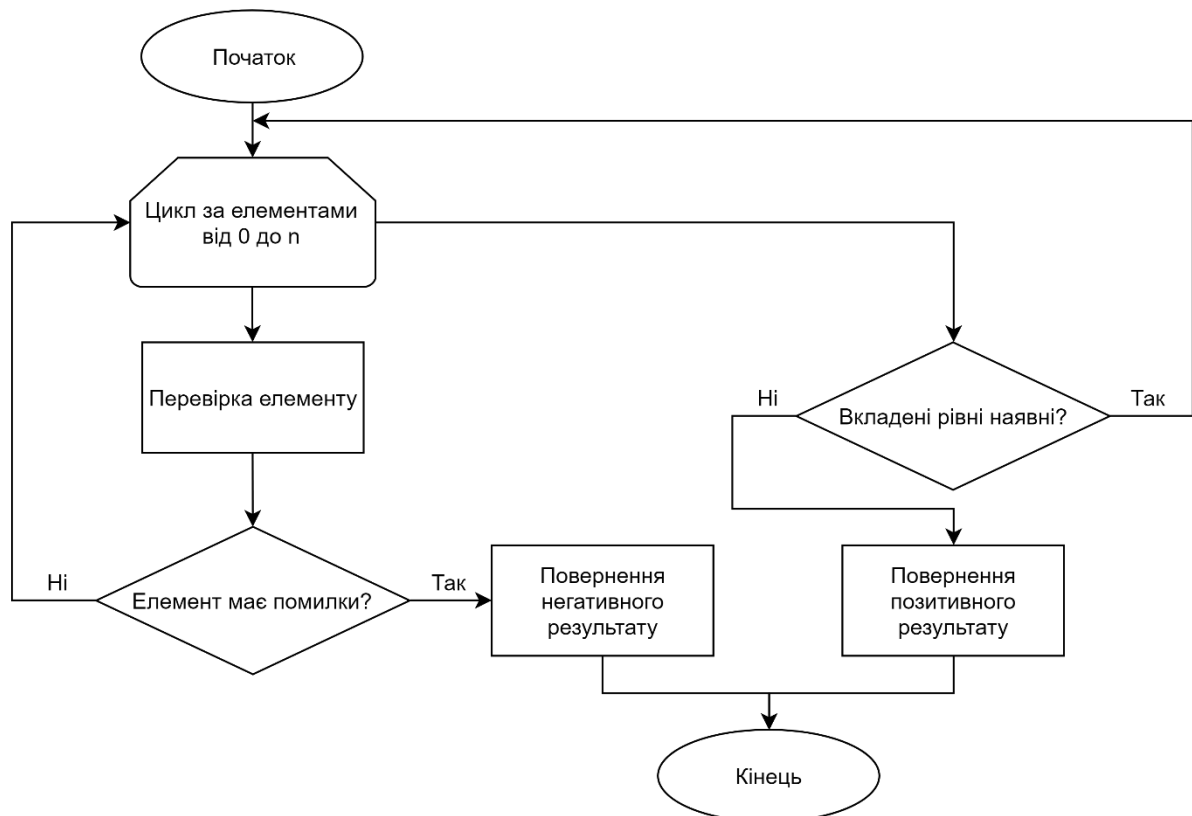


Рис. 2 – Блок-схема аналізуючого алгоритму

2.3.2 Ідентифікуючий модуль

Ідентифікуючий модуль однозначно трансформує виконану користувачку дію у структуру даних, що описує всю інформацію про отриману дію у форматі, зрозумілому для подальшої обробки прогнозуючим модулем.

Алгоритм ідентифікації полягає у послідовній побудові об'єкта, що описує користувачку дію, основуючись на даних, отриманих від графічної

частини програмного забезпечення. У випадку, коли дія, що надійшла, відома алгоритму, їй присвоюється ідентифікатор та починається циклічне зчитування додаткової інформації. Отриманий в результаті об'єкт передається до прогнозуючого модуля. Якщо дія, що надійшла до алгоритму, є невідомою, то система повідомляє про це помилкою в лог подій та нічого не повертає, чим перериває подальшу обробку дії.

Додатково слід зазначити, що інформація стосовно всіх можливих користувацьких дій в програмному застосунку передається ідентифікуючому модулю під час завантаження веб-додатку.

Перетворення опису користувацької дії с наявними додатковими атрибутами в корисну функціональність відбувається таки же чином, лише в протилежному напрямку.

Детальну блок-схему алгоритму ідентифікації зазначено на рис. 3.

2.3.3 Прогнозуючий модуль

Загальне призначення прогнозуючого модуля – побудова списку прогнозованих дій, який утворюється на основі декількох останніх виконаних користувацьких діях та дереві, що описує взаємозв'язки між виконаними діями.

Для своєї роботи прогнозуючий модуль використовує дві основні структури, що зберігають дані: ланцюг користувацьких дій та дерево прецедентів.

Ланцюг користувацьких дій – це послідовність описів користувацьких дій, що зберігається в одновимірному масиві по принципу черги (перший прийшов, перший пішов). Іншими словами можна сказати, що це обмежена за кількістю дій історія користувацької активності.

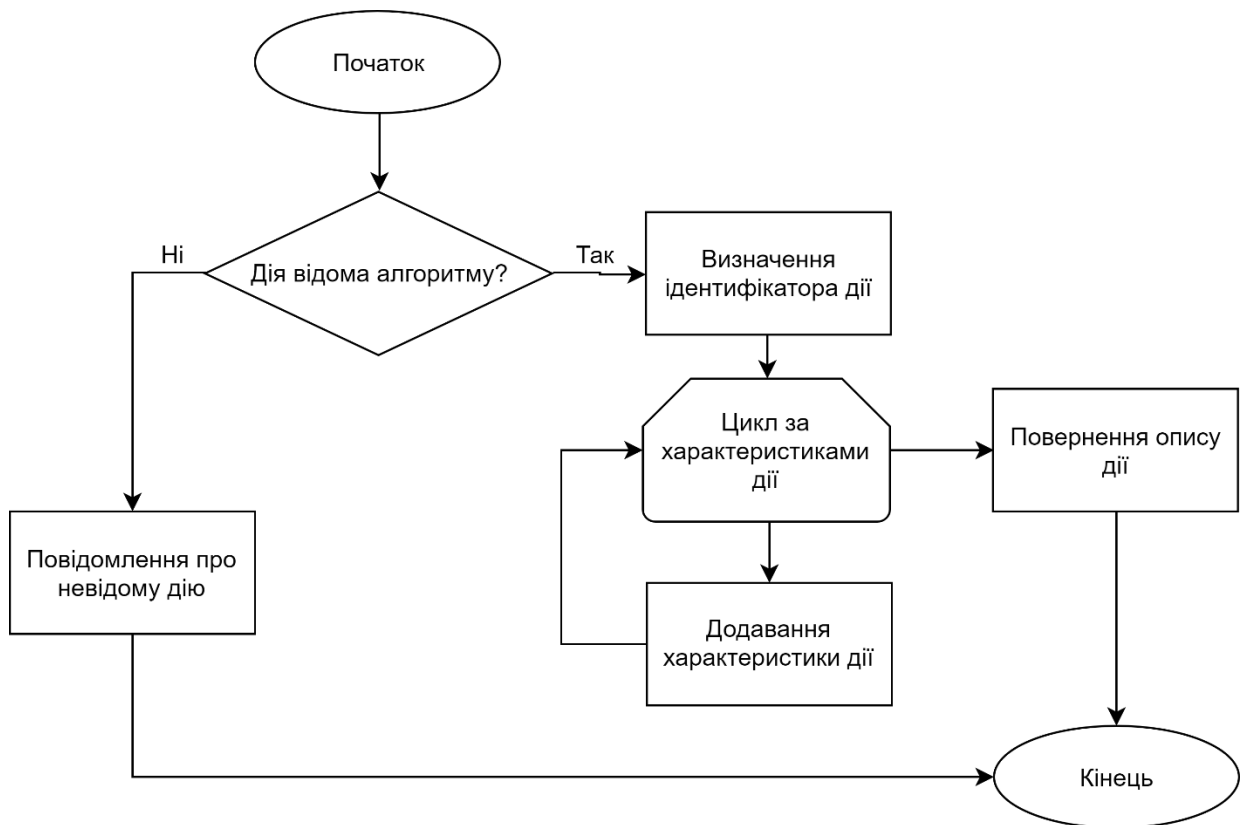


Рис. 3 – Блок-схема ідентифікуючого алгоритму

Дерево прецедентів – N-арне дерево, що зберігає в собі інформацію про кожну виконану дію, включаючи інформацію про додаткові атрибути, та послідовності виконання користувацьких дій протягом всієї роботи із програмним застосунком, доповнені інформацією про повторення окремих.

Приклади ланцюга та дерева прецедентів зображено на рис. 4 - 5.

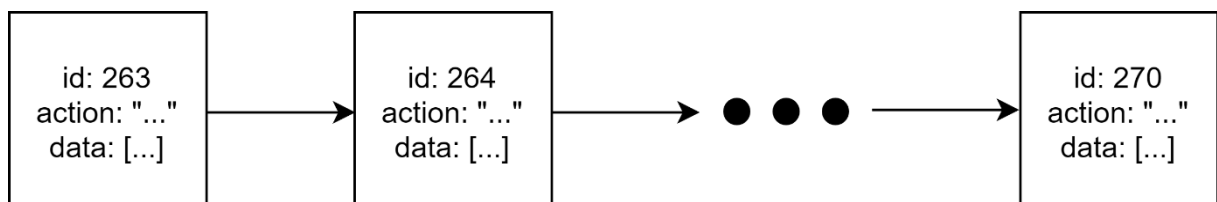


Рис. 4 – Приклад ланцюга користувацьких дій

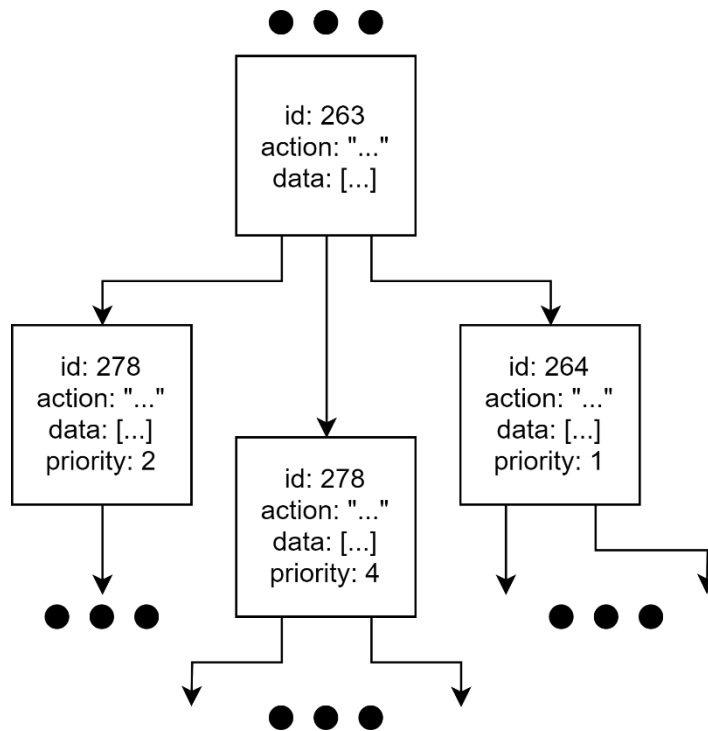


Рис. 5 – Приклад дерева прецедентів

Для побудови списку вірогідних для виконання користувачем дій, прогнозуючий модуль виконує комплекс робіт, а саме:

- редагування ланцюга користувацьких дій;
- побудова дерева прецедентів з описів користувацьких дій;
- пошук входжень елементів в дереві прецедентів.

Редагування ланцюга користувацьких дій призначене для оновлення інформації стосовно актуальних дій, що були виконані. Під час редагування в кінець масиву додається елемент, що описує останню користувацьку дію, а також, з початку масиву видаляється перший елемент, як неактуальний для прогнозування.

Побудова дерева прецедентів з описів користувацьких дій виконується після кожної користувацької активності, що призвела до запису інформації в ланцюг користувацьких дій. Дерево прецедентів будується для

того, щоб побудувати детальну мапу користувацької активності та використовувати її для прогнозування майбутніх користувацьких дій.

Алгоритм побудови дерева прецедентів базується на пошуку входжень перших $N - 1$ елементів з ланцюга користувацьких дій. У випадку, коли входження всього ланцюга знайти не виходить, відсутні елементи ланцюга додаються до дерева прецедентів. Після того, як ланцюг було знайдено, до останнього елемента додається лист, що зберігає в собі інформацію стосовно останньої виконаної користувацької дії, а саме елемент ланцюга на позиції N . Можливий випадок, коли остання виконана дія вже існує на зазначеному рівні, тоді існуючій дії збільшується пріоритет на одиницю.

Блок-схема зазначеного алгоритму будування дерева прецедентів зображена на рис. 6.

Після того, як було завершено побудову дерева прецедентів із додаванням в нього актуальної інформації про останню виконану дію, прогнозуючим модулем виконується процедура пошуку входжень в дереві прецедентів, іншими словами, побудова списку вірогідних до виконання дій.

Для побудови зазначеного списку прогнозуючим модулем запускається пошук входжень ланцюга користувацьких дій в дереві прецедентів. Після того, як входження було знайдене, елементи, що знаходяться на останньому знайденому рівні і утворюють список вірогідних до виконання дій. У випадку, коли входження ланцюга знайдено не було, прогнозуючий модуль повертає порожній список, який символізує недостатню кількість інформації для прогнозування.

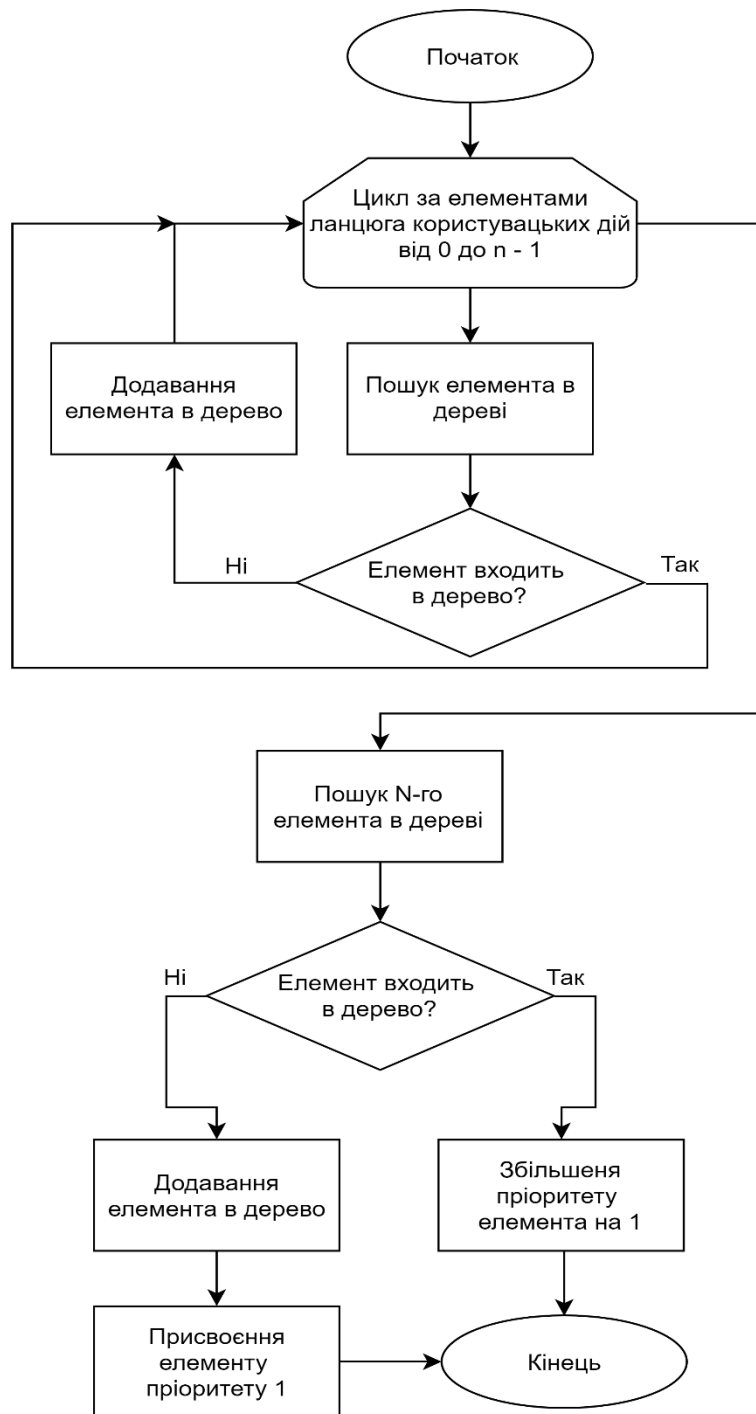


Рис. 6 – Блок-схема алгоритму побудови дерева прецедентів

Список вірогідних до використання користувачем дій сортується згідно с пріоритетом кожної дії, який визначається кількістю повторень конкретної дії. У випадку, коли пріоритети декількох дій мають однакове значення, утворюється колізія пріоритетів. Для вирішення зазначеної колізії використовується порівняння кожної з дій, що призвели до колізії, згідно з

ланцюгом користувацьких дій. Дія, яка має найбільшу кількість пересічень значень атрибутів із діями з ланцюгу, отримує найбільший пріоритет. Відповідно, дія, яка має найменшу кількість пересічень, отримує найменший пріоритет. У випадку, коли кількість пересічень однакова, діям задається однаковий пріоритет.

Детальну блок-схему алгоритму побудови списку вірогідних до виконання дій наведено на рис. 7.

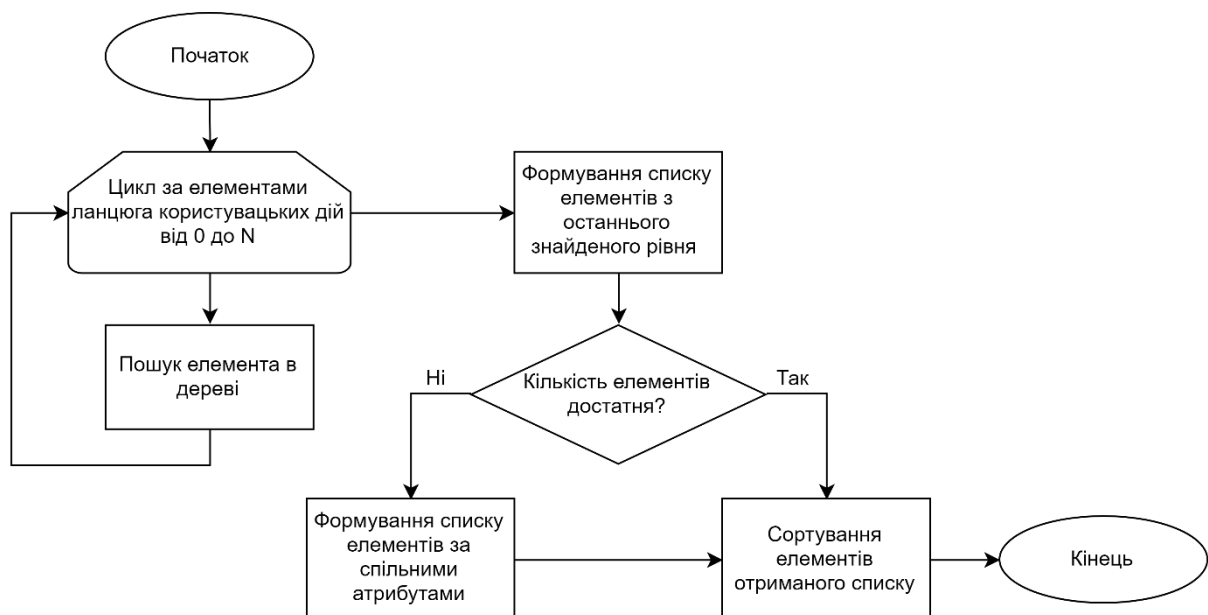


Рис. 7 – Блок-схема алгоритму побудови списку вірогідних до виконання дій

Після того, як прогнозуючим модулем було виконано комплекс зазначених дій, в результаті якого побудовано список прогнозованих дій, наведений список повертається до ідентифікаційного модуля для трансформування його на корисну функціональність, після чого відображається користувачеві для виконання.

2.4 Висновки

Беручи до уваги користувацькі вимоги, що були висунуті до способу та програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва, можна зазначити що спосіб прогнозування є унікальним та повністю дозволяє реалізувати всі поставлені до нього вимоги, а саме: використання власної активності користувача для побудови майбутніх прогнозів та висока точність прогнозів.

Загальне призначення прогнозуючого модуля – побудова списку прогнозованих дій, який утворюється на основі декількох останніх виконаних користувацьких діях та дереві, що описує взаємозв'язки між виконаними діями.

Для своєї роботи прогнозуючий модуль використовує дві основні структури, що зберігають дані: ланцюг користувацьких дій та дерево прецедентів.

Дерево прецедентів – N-арне дерево, що зберігає в собі інформацію про кожну виконану дію, включаючи інформацію про додаткові атрибути, та послідовності виконання користувацьких дій протягом всієї роботи із програмним застосунком, доповнені інформацією про повторення окремих.

Прогнозуючий модуль після виконання будь-якої дії моделювання схеми бізнес-процесів повинен повертати користувачеві прогнозований список подальших дій. Прогнозування вірогідних дій повинно вираховуватися на основі дерева прецедентів, що будується під час всього часу роботи із програмним забезпеченням на основі описів виконаних користувацьких дій, та послідовності декількох останніх дій.

Стосовно програмного забезпечення можна зазначити, що його архітектура відповідає поставленій задачі, через що програмне забезпечення

відповідає вимогам швидкодії та економічності у використанні машинних ресурсів.

3 ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ АЛГОРИТМІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ

3.1 Архітектура програмного забезпечення

Поняття «архітектура» в контексті даного розділу використовується як загальне і може бути поділено на наступні:

- архітектура взаємодії складових частин;
- архітектура програмного забезпечення.

3.1.1 Архітектура взаємодії складових частин

Програмний застосунок, що розроблюється, поділено на дві основні частини: графічна та прогнозуюча. Зазначені частини є повністю незалежними одна від іншої, тобто, у випадку, коли необхідно буде, наприклад, деактивувати одну з них, інша зможе працювати в штатному режимі.

Загалом, модульна архітектура складових частин програмного застосунку прогнозування користувацьких дій при моделюванні схем бізнес-процесів була обрана через її мобільність, легкість в тестуванні та розширенні. Ключовим фактором вибору даного типу архітектури стала саме легкість в розширенні, адже з кожним кроком виконання дослідження з'ясовуються нові обставини, що можуть позитивно впливати на роботу прогнозуючого модуля [15]. При кожному виявленні такого фактору необхідно розширювати логіку алгоритму, що, за допомогою модульної архітектури частин програмного забезпечення, робиться доволі швидко та з малою кількістю помилок. Звичайно, легкість в тестуванні також відіграє велику роль в виборі даного типу архітектури, адже протестувавши працездатність один раз можна бути впевненим, що в подальшій роботі не буде виключних ситуацій, які призведуть до відказу.

Поєднання графічної та прогнозуючої частин програмного забезпечення між собою виконано за допомогою програмного інтерфейсу, що описує просту структуру обміну даних за допомогою публічних методів кожної з частин.

Зображення архітектури взаємодії складових частин програмного забезпечення наведено на рис. 8.

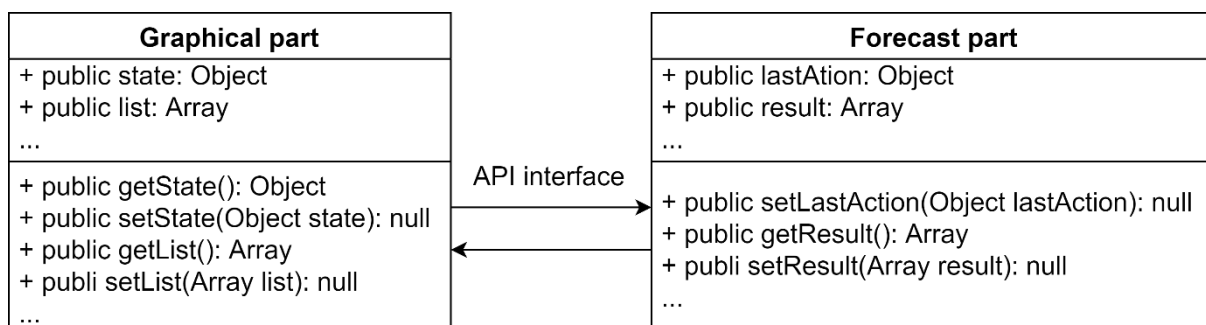


Рис. 8 – Архітектура складових частин програмного забезпечення

3.1.2 Архітектура програмного забезпечення

Загалом, програмне забезпечення, що розроблюється, є клієнт-серверним веб-додатком та складається з двох частин: клієнтської та серверної.

Ключовою особливістю архітектури даного додатку в рамках дослідження, що проводиться, є реалізація всієї необхідної функціональності на клієнтській стороні та відсутність обчислень на серверній стороні за непотрібністю.

Рішення стосовно вибору такого типу архітектури є нестандартним для веб-додатків, але на користь даного рішення вплинули наступні фактори:

- великий розмір інформації, що збирається на клієнтській частині, впливає на великі затрати часу на спілкування з серверною частиною;
- складність організації багатопоточної обробки дерева прецедентів;
- надлишкова складність організації оптимізованого зберігання дерева прецедентів в базі даних.

Підсумувавши зазначені фактори, можна зробити висновок, що розташування прогнозуючого модуля на серверній частині веб-додатку призвело би до надлишкових затрат часу на прогнозування користувацьких дій та організацію самої серверної частини. Набагато простішим та швидшим рішенням, в рамках дослідження, що проводиться, є розташування прогнозуючого модуля та організація зберігання користувацьких даних на клієнтській частині веб-додатку та надання серверній частині ролі файлового серверу, що зберігає статичні файли веб-додатку.

Описана архітектура зображена на рис. 9.

3.2 Опис інтерфейсу веб-додатку

Розроблений програмний додаток та графічний інтерфейс до нього буде доповнено графічним модулем, що буде демонструвати користувачеві доступні до виконання дії, які є результатом роботи прогнозуючого модуля на кожному кроці конструювання схеми бізнес-процесів. Елементи зазначеного модуля будуть доступні до виконання, про що користувачу буде повідомлено за допомогою стильових особливостей, які змінюються при

наведенні курсором миші на елементи. Також, присутню панель управління елементами, буде доповнено кнопкою повернення до попереднього стану схеми бізнес-процесів, реалізація якої буде базуватися на збереженні історії виконаних користувацьких дій та відповідних станів схеми бізнес-процесів, що конструюється.

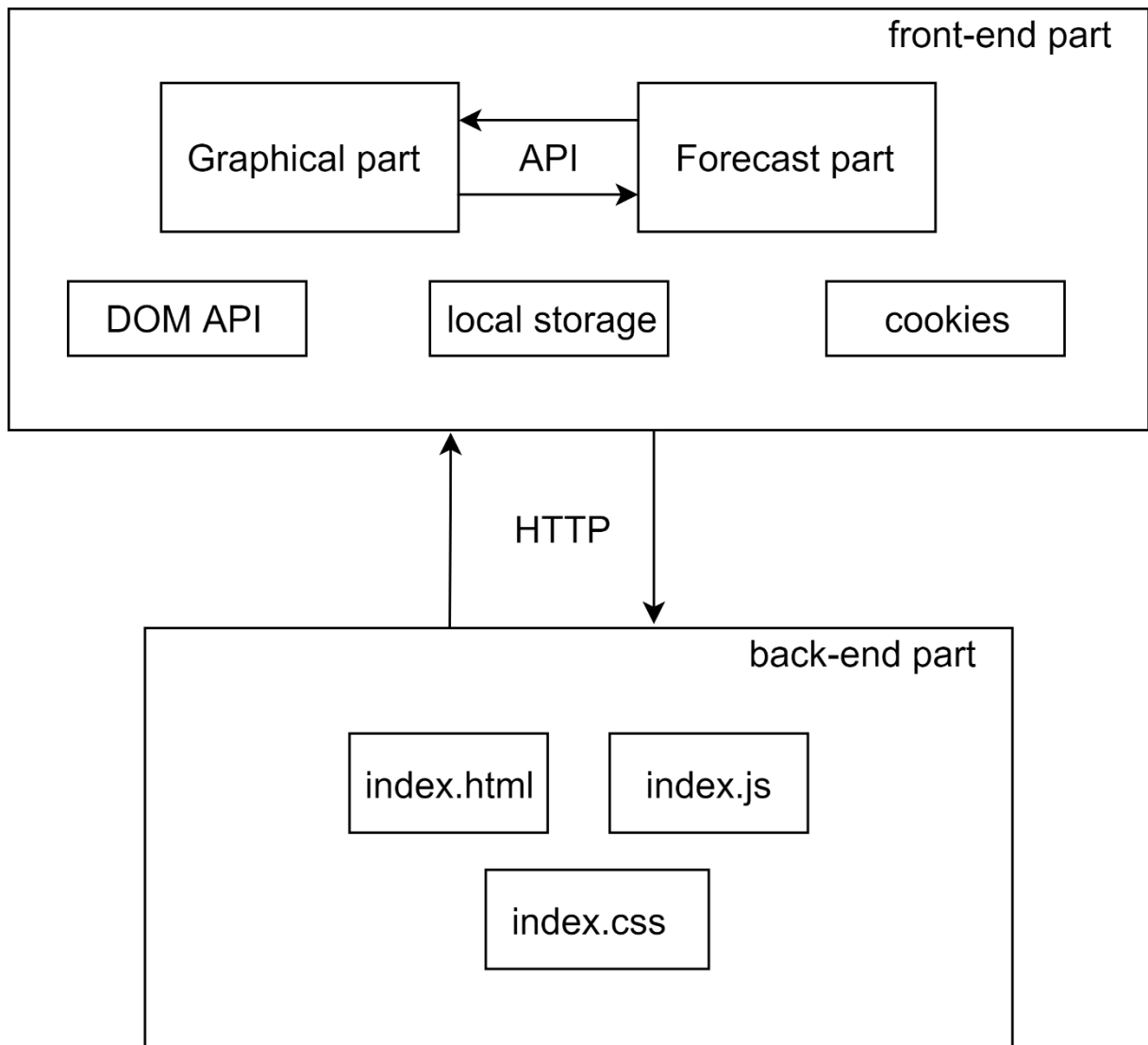


Рис. 9 – Загальна архітектура веб-додатку

Структуру блоку, що зберігає запропоновані дії для виконання користувачем, зазначено на рис. 10.


- Додати блок "Робота" до схеми
- Додати вхідні дані для створеного блоку "Робота"
- Затунелювати результат блоку "Робота" з наступним рівнем схеми 

Рис. 10 – Структура блоку запропонованих для виконання дії

3.3 Засоби та технології реалізації програмного забезпечення

Оскільки додаток, що розроблюється, виконується на платформі веб, то для його реалізації необхідно окремо організувати роботу клієнтської та серверної частин.

Серверна частина буде реалізована на основі серверу Nginx, головною задачею якого буде повернення статичних файлів програмного забезпечення, що виконуються в клієнтському браузері.

Клієнтська частина буде реалізована на мові програмування JavaScript версії ES6 з підключеною бібліотекою ReactJS для створення інтерфейсу користувача. До клієнтської частини також буде застосована технологія збірки та мініфікації початкових файлів, для реалізації якої клієнтська частина буде використовувати збірник Webpack.

До додаткових кейсів використання збірника Webpack можна віднести наступні:

- використання модуля Webpack-Dev-Server як локального серверу під час розробки програмного забезпечення;
- трансліювання сучасної версії мови JavaScript ES6 в більш стару ES5 задля збільшення кількості браузерів, що підтримуються для виконання програмного забезпечення.

Для повноцінного функціонування програмного забезпечення, що розроблюється, також необхідне використання додаткових технологій, що надаються користувацьким браузером, а саме:

- локальне сховище даних – LocalStorage API;
- інтерфейс використання додаткових потоків виконання програмних процедур – WebWorker API.

3.4 Вимоги до платформи виконання програмного забезпечення

Головними вимогами для користувацьких браузерів, що будуть виконувати програмне забезпечення, є:

- можливість виконання мови програмування JavaScript версії ES5 та вище;
- надання локального сховища для зберігання користувацьких даних LocalStorage API;
- підтримка сучасного стандарту HTML5, що включає в себе функціональність WebWorker API.

В таблиці 1 представлено сумісність перелічених технологій з версіями сучасних браузерів.

Таблиця 1. Сумісність технологій із версіями браузерів

	Google Chrome	Internet Explorer	Mozilla Firefox	Opera	Safari
JavaScript ES5 та вище	усі	усі	усі	усі	усі
LocalStorage API	усі	усі	усі	усі	усі
WebWorker API	4.0	10.0	3.5	10.6	2.0

Виходячи з результатів аналізу, для повноцінного використання функціональності програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва користувачі мають використовувати наступні браузери:

- Google Chrome v4.0 та вище;
- Internet Explorer v10.0 та вище;
- Mozilla Firefox v3.5 та вище;
- Opera v10.6 та вище;
- Safari v2.0 та вище.

3.5 Особливості реалізації головних алгоритмів та структур даних

3.5.1 Аналіз схеми бізнес-процесів на наявність помилок

Оскільки аналіз схеми бізнес-процесів на наявність помилок виконується після кожної зареєстрованої користувацької дії, то, відповідно, його робота повинна займати мінімальну кількість часу, адже цей процес входить в процес прогнозування користувацьких дій і, відповідно, весь затрачений прогнозуючим модулем час користувач знаходиться в очікуванні результатів обробки.

Для вирішення зазначеної проблеми я використовував складний вкладений об'єкт, що складається з описів елементів, які входять до конкретного рівня сконструйованої схеми бізнес-процесів та включає в себе описи всіх дочірніх рівнів з, відповідно, наявними в них елементами.

Такий формат структури даних надає можливість перевіряти кожний з елементів, що знаходиться на конкретному рівні схеми бізнес-процесів на наявність помилок та, за допомогою зберігання деяких властивостей елемента в глобальному середовищі відносно заданого елемента, перевіряти на наявність помилок залежні від конкретного елементи.

Таким чином, використовуючи рекурсивний алгоритм перебору об'єкту, що описує схему бізнес-процесів, я отримав можливість за малий проміжок часу виконувати перевірку всієї схеми бізнес-процесів на наявність помилок та, у разі допущення користувачем помилки, вказувати конкретний рівень вкладеності схеми бізнес-процесів та елемент, що належить цьому рівні вкладеності, який є помилковим.

На рис. 11 наведено детальну структуру багаторівневого об'єкту, що описує схему бізнес-процесів виробництва.

3.5.2 Опис користувацької дії в ідентифікуючому модулі

Основна задача ідентифікуючого модуля – це трансформування початкової інформації про виконану користувачем дію в опис виконаної дії, що має певний формат, зрозумілий прогнозуючому модулю, за допомогою атрибутів, які мають структуру «ключ – значення». Виконання такого трансформування необхідне для приведення інформації про будь-яку зареєстровану дію до єдиного вигляду, так як цього вимагає модульна архітектура прогнозуючого модуля.

```

{
  level: 1,
  name: "Root",
  elements: [
    {
      id: 1
      type: "arrow",
      source: 4,
      destination: 10,
      from: "right",
      to: "left",
      title: "Arrow name"
    },
    {
      id: 4,
      type: "block",
      title: "Block name",
    },
    ...
  ],
  sublevels: [
    {
      level: 2,
      name: "Block name"
      elements: [
        ...
      ],
      sublevels: [
        ...
      ]
    }
  ]
}

```

Рис. 11 – Структура об'єкту, що описує схему бізнес-процесів

У випадку, коли ідентифікація кожної зареєстрованої дії не проводилася би, розширення функціональності прогнозуючого модуля відбувалося б по сценарію «вивчення» особливостей кожної нової дії та написання алгоритму її обробки. Такий підхід при розробці програмного забезпечення вважається поганою практикою, адже він збільшує зв'язність програмних компонент між собою, ускладнює код програми для подальшої підтримки та збільшує час роботи модуля.

Для того, щоб реалізувати повноцінну функціональність ідентифікації користувацьких дій, модуль, під час завантаження програмного застосунку,

отримує список усіх можливих дій, які можуть бути зареєстровані в програмному застосунку, та генерує з цього списку набір методів публічного інтерфейсу, які надалі передає у глобальне середовище програмного застосунку. Після того, як користувач виконав будь-яку зареєстровану дію, викликається відповідний метод, що надається ідентифікуючим модулем, в який передаються параметри, що описують виконану дію. В ролі параметрів, що передаються, можуть виступати наступні:

- інформація про браузерну подію, що спричинила виклик методу інтерфейсу модуля;
- часовий проміжок, коли була виконана дія;
- дані про стан схеми-бізнес процесів на конкретному рівні;
- додаткові атрибути.

Слід зазначити, що в деяких випадках, необхідні атрибути можуть бути описані в початковому списку, за яким генерується методи публічного інтерфейсу модуля. Завдяки такому підходу збільшується надійність ідентифікуючого модуля та точність прогнозування, адже всі дані, які прогнозуючий модуль очікує, будуть йому надані.

Після того, як ідентифікуючий модуль отримав інформацію про користувацьку дію, він перевіряє інформацію на відповідність шаблону. У випадку, коли інформація пройшла перевірку, ідентифікуючий модуль вибирає необхідну інформації із початкових складних об'єктів, формує новий об'єкт та надає йому унікальний ідентифікатор виконаної дії. Результат ідентифікуючого модуля передається та оброблюється прогнозуючим модулем.

Лістинг програмного коду, що реалізує ідентифікуючий модуль, наведено в таблиці 2.

Таблиця 2. Лістинг програмного коду ідентифікуючого модуля

Identifier.js
<pre>// опис класу Identifier та реалізація основних методів class Identifier { constructor(data) { this.keys = Object.keys(data); Object.entries(data).forEach(([key, value]) => { this[key] = function () { const args = Array.from(arguments); const attributes = value.reduce((res, val, index) => { return res !== undefined && val(args[index]) ? Object.assign({}, res, {[val.name]: args[index]}) : undefined }, {}); if (attributes) { Forecast.push({ id: key, attributes }); } }; }); } getPublicMethods() { return this.keys.reduce((res, key) => { return Object.assign({}, res, {[key]: this[key]}) }, {}) } }; const IdentifierAttributes = { event: (attr) => typeof attr === "string", date: (attr) => typeof attr === "number" }; ... // передача класу Identifier дій, що можуть зустрітися в додатку // та формування публічних методів</pre>

```
const methods = new Identifier({
  action: [
    IdentifierAttributes.event,
    IdentifierAttributes.date
  ]
}).getPublicMethods();

export const {...methods};
```

3.5.3 Особливості реалізації дерева прецедентів

Дерево, що буде використовуватися для вирішення задачі прогнозування, матиме наступну структуру. Кореневим елементом дерева слугуватиме спеціальний об'єкт, що вказує на місце початку пошуку в дереві. Глибина вкладеності дерева, тобто кількість рівнів елементів, буде рівнятися довжині ланцюгу користувацьких задач. На рівнях від першого до передостаннього зберігатимуться елементи, що мають навігаційну роль в процесі побудови списку можливих до виконання дій, тобто за допомогою цих елементів відбуватиметься пошук останнього рівня дерева прецедентів, на якому будуть зосереджені описи користувацьких дій, можливі до виконання на останньому кроці користувацької активності.

В рамках даної задачі для досягнення найбільшої продуктивності при виконанні додавання нового або пошуку існуючого елемента необхідно використовувати пошук по дереву в ширину, адже це надає можливість пришвидшення виконання операцій за рахунок ігнорування сторонніх гілок дерева, які не входять в актуальний маршрут користувацьких дій.

Додатковою особливістю дерева прецедентів є економія пам'яті у випадках, коли ланцюг актуальних користувацьких дій вказує на вже існуючу дію, що зберігається в дереві прецедентів. В такому випадку, замість додавання нового елемента на конкретний рівень, в існуючому елементі збільшується пріоритет на одиницю та додаються додаткові

значення атрибутів, що описують конкретну користувацьку дію. В результаті, модифікований елемент отримує більший пріоритет відносно інших елементів даного рівня і в подальших прогнозах займатиме вищу позицію в побудованому списку вірогідних до виконання користувацьких дій.

В таблиці 3 наведено лістинг коду, що реалізує основну функціональність дерева прецедентів.

Таблиця 3. Лістинг реалізації дерева прецедентів

Tree.js
<pre>class Tree { constructor() { this.root = null; } makeNode(value) { return { value, left: null, right: null }; }; add(value) { const currentNode = this.makeNode(value); if (!this.root) { this.root = currentNode; } else { this.insert(currentNode); } return this; }; insert(currentNode) { const value = currentNode.value; const traverse = function(node) { if (value > node.value) { if (!node.right) { node.right = currentNode; return; } } }; traverse(this.root); } }</pre>

```

        } else traverse(node.right);
    } else if (value < node.value) {
        if (!node.left) {
            node.left = currentNode;
            return;
        } else traverse(node.left);
    }
};

traverse(this.root);
};

find(value) {
    const node = this.root;
    const traverse = function(node) {
        if (!node) return false;
        if (value === node.value) {
            return true;
        } else if (value > node.value) {
            return traverse(node.right);
        } else if (value < node.value) {
            return traverse(node.left);
        }
    };
    return traverse(node);
};

getDepth() {
    const node = this.root;
    let maxDepth = 0;
    const traverse = function(node, depth) {
        if (!node) return null;
        if (node) {
            maxDepth = depth > maxDepth ? depth : maxDepth;
            traverse(node.left, depth + 1);
            traverse(node.right, depth + 1);
        }
    };
    traverse(node, 0);
    return maxDepth;
};

```

3.5.4 Особливості реалізації графічного інтерфейсу

Графічним інтерфейсом в даній роботі виступає компонент, який надає можливості відображення та взаємодії зі списком прогнозованих дій. Таким чином, для реалізації зазначеного модуля необхідно реалізувати графічний компонент за допомогою бібліотеки ReactJS, який при виборі користувача необхідної прогнозованої дії, буде викликати публічний метод прогнозуючого модуля.

Компонент складається з наступних частин:

- обгортка, що демонструє компонент загалом;
- список подій, що отримує функціональність та розподіляє її між елементами списку;
- елемент списку подій, що отримує та викликає передану функціональність.

За допомогою методів життєвого циклу, компонент підписаний на зміні загального стану веб-додатку, що дозволяє йому при кожній зміні своєчасно оновлювати актуальні дані про список прогнозованих дій.

На рис. 12 зображено структуру компоненту з урахуванням складових частин.

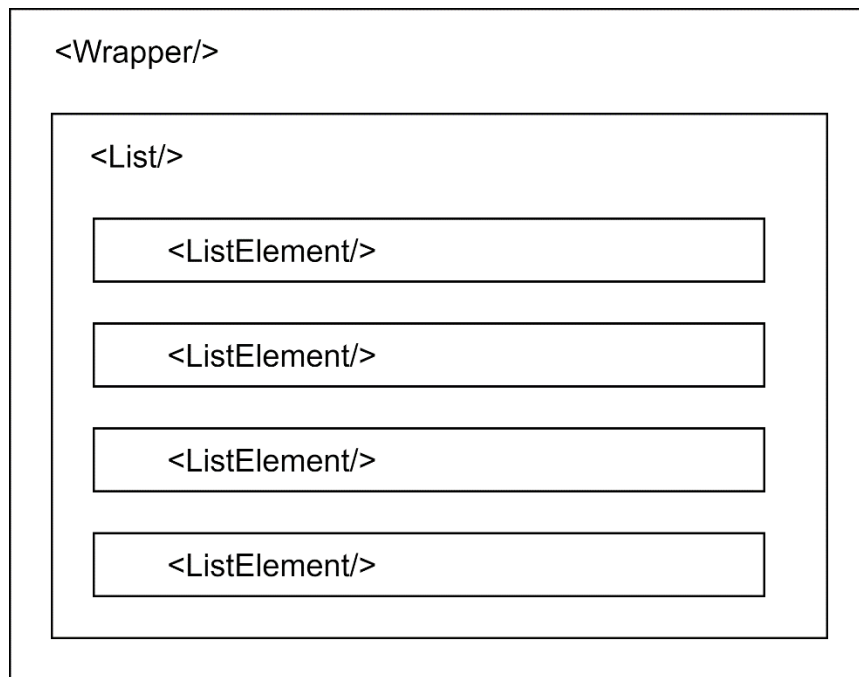


Рис. 12 – Структура компоненту

3.6 Висновки

Для реалізації програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва буде використано клієнт-серверну архітектуру для додатку на платформі веб, що складається з двох основних частин, графічної та прогнозуючої, побудованих за модульною архітектурою та поєднаних між собою за допомогою програмного інтерфейсу взаємодії компонент через публічні методи класів.

Вибір даного архітектурного рішення для заданого програмного забезпечення зумовлено наступними факторами:

- реалізація програмного забезпечення прогнозування користувацьких дій відбувається на основі існуючого програмного застосунку на платформі веб;
- окремі частини програмного забезпечення прогнозування користувацьких дій повинні бути незалежними одна від іншої;

- вибрана архітектура повинна надавати можливості легкого розширення функціональності програмного забезпечення, що розроблюється.

Оскільки досягнення оптимальних показників продуктивності прогнозуючого модуля з малими витратами машинної пам'яті було важливою задачею, що ставилась до розроблення програмного забезпечення, то використання описаних особливостей було необхідним рішенням.

Також слід зазначити, що перевіривши працездатність програмного застосунку на практиці, спостерігався позитивний вплив на швидкодію веб-додатку.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Тестування розробленого програмного забезпечення

Тестування розробленого програмного забезпечення виконувалося в декілька етапів в процесі реалізації програмного забезпечення, а саме:

- тестування розробленої функціональності прогнозування користувацьких дій методом білого ящика;
- тестування розробленого користувацького інтерфейсу програмного забезпечення методом чорного ящика.

Проаналізувавши основні методи тестування програмного забезпечення, що має користувацький інтерфейс, я вирішив проводити тестування графічної та прогнозуючої частин програмного забезпечення окремо, адже для кожної з них існують окремі вимоги та стандарти тестування [17].

Для прогнозуючої частини я обрав метод білого ящика, адже цей метод є дуже ефективним для знаходження багів у програмному коді у випадку, коли тестувальнику повністю відома структура коду. Оскільки програмне забезпечення з самого початку розроблене мною, то виконання тестування не зайняло великої кількості часу і за допомогою цього методу було знайдено і виправлено декілька важливих для точності прогнозування багів.

Для тестування графічної частини я обрав метод чорного ящика через його особливість відсторонення від реалізації програмного коду і концентрації уваги на тестуванні саме якості роботи програмного забезпечення з точки зору кінцевого користувача. За допомогою цього методу мною було знайдено один важливий баг в користувацькому

інтерфейсі, який я упустив про розробці графічної частини програмного забезпечення.

За результатами проведеного тестування можна відзначити, що реалізоване програмне забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва не має в собі критичних багів, що можуть негативно вплинути на якість виконання програмного забезпечення під час роботи із ним кінцевого користувача.

4.2 Аналіз розробленого програмного застосунку

Після завершення розробки програмного забезпечення прогнозування користувацьких дій мною було виконано аналіз програмного забезпечення на відповідність користувацьким вимогам, описаним в п. 2.1., аналіз якості програмного коду, графічного інтерфейсу та окремо функціональності прогнозуючого модуля.

Розроблене програмне забезпечення:

- надає графічний користувацький інтерфейс для взаємодії із згенерованим списком вірогідних до виконання дій;
- реалізовує функціональність для аналізу побудованої схеми бізнес-процесів на наявність помилок та можливості повернення схеми до попереднього стану;
- в автоматичному режимі будує список можливих до виконання користувачем дій за проміжок часу менший ніж 1000 мілісекунд, що ґрунтується на попередніх виконаних діях.

Таким чином, можна зазначити, що в розробленому програмному забезпеченні усі основні користувацькі вимоги були виконані.

Проаналізувавши програмний код після завершення розробки та виконання тестування розробленого програмного забезпечення можна з впевненістю заявити, що усі «вузькі місця» програмного застосунку мають необхідні перевірки та повністю контролюють усі можливі виключні ситуації, пов'язані з роботою програмного забезпечення.

Аналіз якості графічного інтерфейсу також показав гарний результат, адже програмний застосунок, що виконується на платформі веб, має дружній до користувача інтерфейс, який включає в себе інтерактивні елементи, які явно та неявно підказують користувачеві як користуватися функціональністю додатку. Також слід відзначити велику швидкість реагування програмного застосунку на користувацькі дії та оновлення даних, що також додає комфорту та ясності при роботі із додатком.

Окремо слід навести аналіз прогнозуючого модуля програмного забезпечення, а саме залежність точності та часу, необхідного для його роботи, від довжини ланцюга користувацьких дій. За результатами проведеного аналізу можна побудувати модель залежності, графіки якої наведено на рис. 12 - 13.

Як видно з першого графіку, час побудови, як і очікувалось, зростає експоненціально, залежно від кількості елементів, що входять в ланцюг останніх користувацьких дій.

Особливу увагу слід приділити фрагменту другого графіку між точками на 6 та 10 на осі довжини ланцюгу користувацьких дій. Точність прогнозів в цьому місці відповідає поставленим вимогам та змінюється незначно. Тому, в використанні більшої довжини ланцюгу користувацьких дій немає сенсу.

Таким чином, виходячи із аналізу двох графіків, можна сказати, що оптимальною довжиною ланцюгу користувацьких дій є 6 елементів.

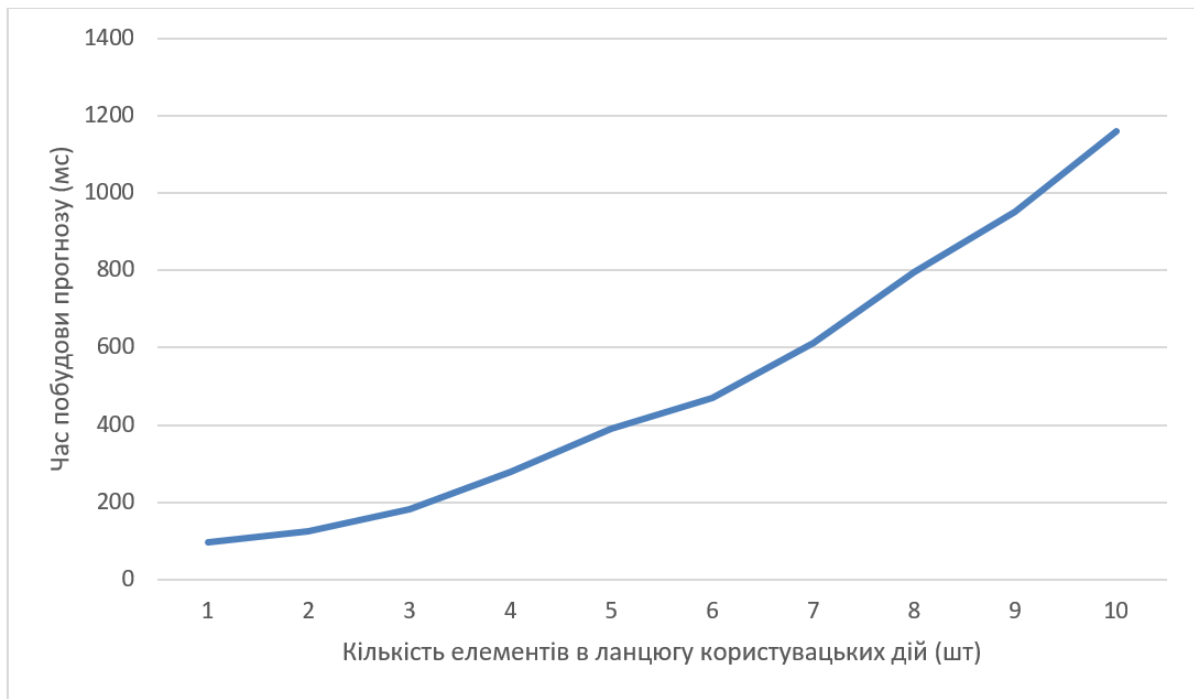


Рис. 12 – Графік залежності часу побудови прогнозу від кількості елементів в ланцюгу користувацьких дій

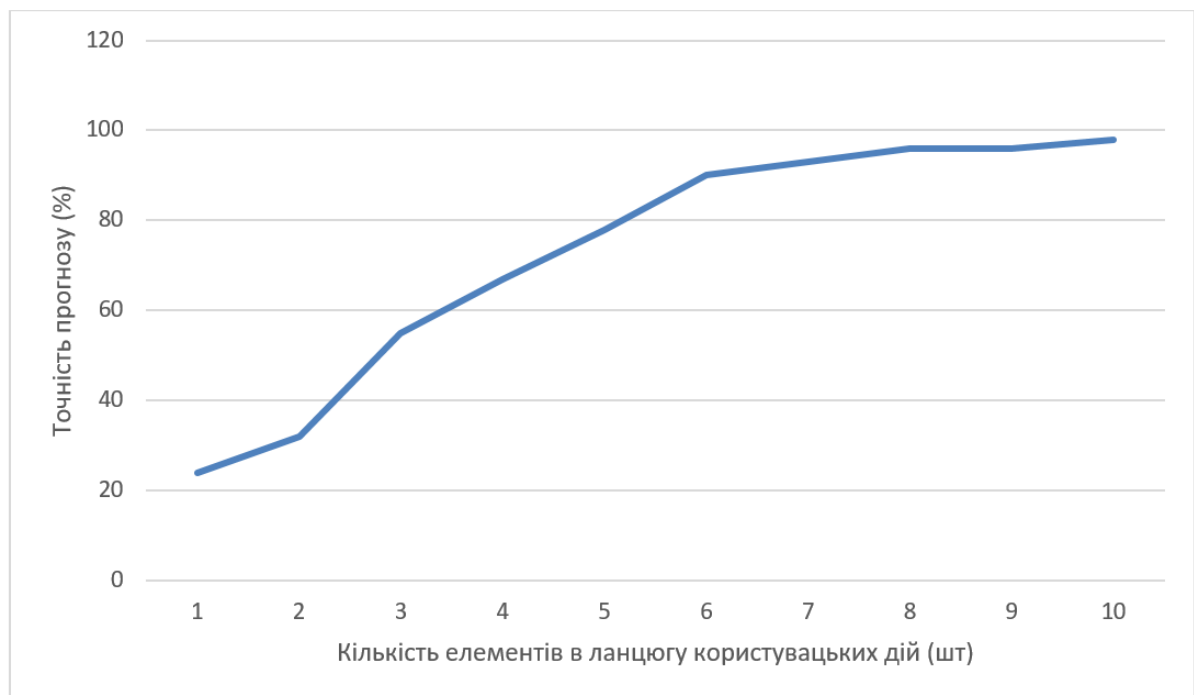


Рис. 13 – Графік залежності точності прогнозу від кількості елементів в ланцюгу користувацьких дій

4.3 Вдосконалення розробленого програмного забезпечення

В рамках даної роботи розроблене програмне забезпечення прогнозування користувацьких дій повністю виконує назначені йому функції, але, все ж таки, за результатами аналізу кінцевого програмного застосунку можна виділити деякі моменти, які можна оптимізувати з метою пришвидшення роботи програмного забезпечення, зменшення машинної пам'яті, необхідної для роботи застосунку, та покращення результатів прогнозування користувацьких дій.

До таких вдосконалень можна віднести наступні:

- реалізація динамічної довжини ланцюгу користувацьких дій та, відповідно, динамічної максимальної вкладеності дерева прецедентів;
- використання події повернення до попереднього стану для зменшення пріоритету обраної дії, після якої виконувалося повернення;
- можливість демонстрації результатів прогнозованої дії без необхідності її виконувати.

Наведемо більш детальний аналіз кожного з вдосконалень програмного забезпечення.

4.3.1 Динамічна довжина ланцюгу користувацьких дій

На даний момент в розробленому програмному забезпеченні, а саме в прогнозуючому модулі, існує умова, яка обмежує довжину ланцюгу користувацьких дій і, відповідно, глибину вкладеності дерева прецедентів. Відповідно до аналізу, наведеного в п. 4.2, ця умова обмежує кількість елементів у кількості 6 одиниць, адже в цьому випадку прогнозуючий модуль є найбільш ефективним. Але цей показник набуває актуальності у випадку, коли дерево прецедентів достатньо заповнено елементами для

виконання прогнозів, які повертають від 3-х до 5-ти вірогідних для виконання дій, що є найбільш зручним для кінцевого користувача.

У випадку, коли даних в дереві прецедентів недостатньо, тобто на стадії початкового навчання програмного забезпечення, через зазначену довжину ланцюга користувацьких дій модуль прогнозування вірогідних дій може доволі довго набирати оберти для повернення зручного для користувача прогнозу. Це не є помилкою в роботі програмного застосунку, а є його особливості.

Таким чином, реалізувавши динамічну довжину ланцюгу користувацьких дій, яка залежить від кількості елементів, що вже зібрані та структуровані прогнозуючим модулем, можна досягти повернення користувачеві зручного набору вірогідних до виконання дій вже після виконання 20 – 25 дій із програмним забезпеченням, тобто після 2-3 хвилин після початку роботи.

4.3.2 Зменшення пріоритету дії

Проаналізувавши специфіку роботи із розробленим програмним забезпеченням, можна зробити висновок, що подія повернення до попереднього стану сконструйованої схеми бізнес-процесів виробництва є виключною в контексті інших подій, що користувач може виконати при роботі.

Оскільки результат події відміни можна трактувати як відмова від наведених змін за їх невідповідністю очікуванням, можна зробити висновок, що виконана подія, яку відміняє користувач, була невірною і не підходить для редагування схеми бізнес-процесів на конкретному кроці.

Таким чином, збільшити точність прогнозуючого модуля можна за рахунок опрацювання події повернення до попереднього стану схеми

бізнес-процесів у випадку, коли повернення виконувалося після виконання дії зі списку, побудованого прогнозуючим модулем.

Блок-схему алгоритму опрацювання описаної ситуації наведено на рис. 14.

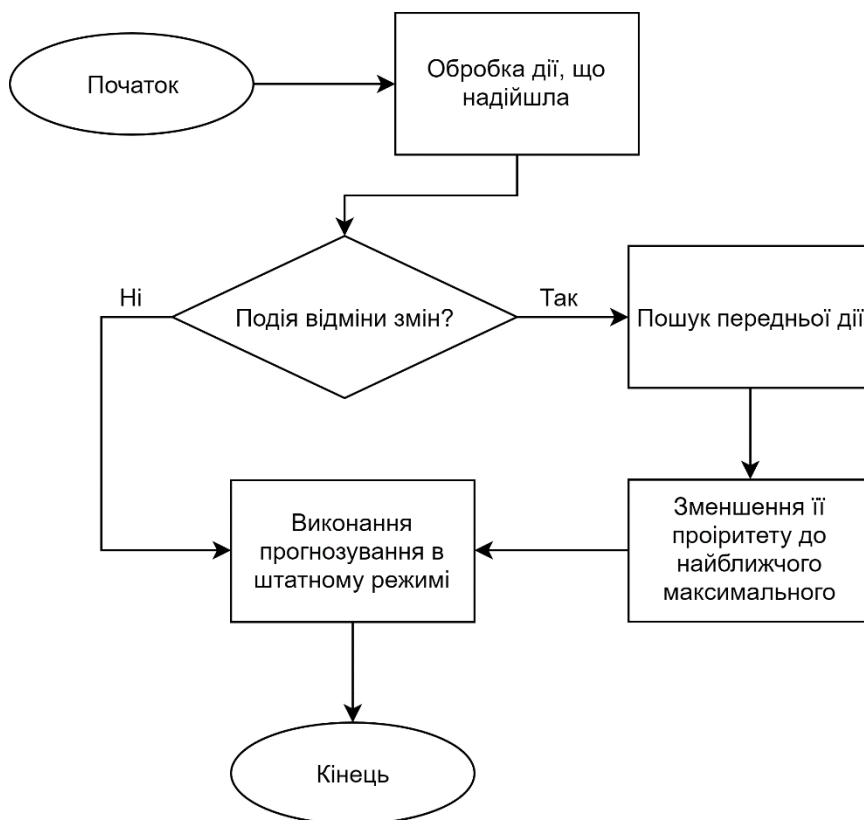


Рис. 14 – Блок-схема алгоритму зменшення пріоритету дії за подією повернення до попереднього стану схеми бізнес-процесів

4.3.3 Демонстрація результату прогнозованої дії

Зазначене вдосконалення відноситься до графічної частини програмного застосунку прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.

Існують деякі випадки, коли опис прогнозованої дії, що відображається користувачеві у графічному модулі програмного

забезпечення, може бути недостатньо інформативним для прийняття рішення до виконання цієї дії, наприклад:

- дуже скорочений за довжиною опис, що не відображає суті конкретної дії;
- відсутність локалізації програмного забезпечення для конкретної країни;
- труднощі перекладу для регіону з особливим діалектом.

В зазначених випадках для того, щоб користувачеві зрозуміти який результат буде після виконання обраної дії, йому необхідно виконати цю дію. Відповідно, якщо результат дії йому не підходить, йому потрібно буде повернутися до попереднього стану схеми бізнес-процесів.

Описаний хід подій є дуже незручним, тому для більшої зручності користувача, до графічної частини програмного забезпечення можна додати функціональність демонстрації результату на етапі, коли користувач наводить курсор на прогнозовану дію, але застосовувати зміни, як і раніше, тільки у випадку натискання кнопки миші на обраній дії.

На мою думку, реалізація зазначеної функціональності допоможе користувачеві в період звикання до програмного забезпечення та зменшить загальний час, необхідний на конструювання схем бізнес-процесів.

4.4 Висновки

За результатами тестування та аналізу кінцевого програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва можна заявити, що розроблений програмний застосунок не має помилок, які можуть призвести до виключних ситуацій при роботі користувача та повністю відповідає початковим вимогам, що

були висунуті в процесі анкетування потенційних користувачів програмного застосунку.

Також було побудовано список можливих до реалізації вдосконалень розробленого програмного застосунку з метою зменшення навантаження на апаратну частину виконуючої машини та збільшення точності результатів процесу прогнозування вірогідних до виконання користувачем дій.

5 ПОБУДОВА БІЗНЕС-МОДЕЛІ

5.1 Опис проблеми

В сфері проектування операційної діяльності на даний момент існують дві основні проблеми:

- наявність семантичних помилок в кінцевих схемах бізнес-процесів;
- надлишкові витрати часових ресурсів, необхідних на проектування виробничої діяльності.

Обидві зазначені проблеми знаходяться на високому рівні пріоритетів для користувача, адже ігнорування зазначених проблем веде к надлишковим затратам матеріальних ресурсів при проектуванні виробничої діяльності.

Для повноти розуміння зазначених проблем побудуємо дерево проблем, на якому зображено конкретні причини та наслідки труднощів, з якими користувач може зустрітися при ігноруванні проблем. Дерево проблем наведено на рисунку А.1.

Вирішивши зазначені проблеми, користувач отримає можливості виявлення семантичних помилок в схемі бізнес-процесів на етапі її побудови та економії часових ресурсів, необхідних на проектування, за допомогою чого зможе економити матеріальні ресурси, виділені на проектування.

На даний момент для користувача не існує єдиного рішення зазначених проблем.

У деяких програмних продуктах розроблені модулі для здійснення фінальних перевірок сконструйованих схем бізнес-процесів на наявність нотаційних помилок (наприклад ErWin, Terrasoft), що є першим існуючим рішенням зазначеної проблеми.

Основний недолік зазначених продуктів полягає в тому, що вони шукають помилки в кінцевій реалізації схеми бізнес-процесів, тобто фактично на етапі, наступному після проектування системи, що призводить до більших затрат на виправлення допущених помилок, і не мають можливості прогнозування виникнення помилок на етапі конструювання схем бізнес-процесів.

Стосовно вирішення проблеми економії часових на даний момент не існує популярних рішень в цій сфері через її специфічність та неоднозначність.

В силу того, що модуль, що надає рішення зазначених проблем, тісно пов'язаний зі специфікою архітектури конструктора схем бізнес-процесів, поставлятися він буде комплексно з вищевказаним конструктором. Для деяких користувачів це може послужити перешкодою при переході з існуючого рішення на запропоноване, але якісна оцінка переваг, що користувач отримує при переході, затьмарить собою незручності, пов'язані з новим інтерфейсом і специфікою роботи з конструктором.

5.2 Зацікавлені сторони

В ході побудови бізнес-моделі мною були визначені наступні зацікавлені в реалізації проекту сторони:

- проектний менеджер;
- інвестор;
- виконавець спроектованої системи;
- конкуренти.

Порівняльний аналіз зацікавленості в проекті, умов зацікавленості в довгостроковості проекту та вірогідної участі наведено в таблиці А.1 в додатку А.

5.3 Опис рішення проблеми

Рішенням зазначеної проблеми, на мою думку, є накопичення і обробка інформації про користувацькі дії та проаналізованих результатів виконання цих дій. Під аналізом результатів мається на увазі аналіз проміжної схеми бізнес-процесів на наявність допущених помилок.

Таким чином, маючи в розпорядженні подібний набір даних, рішення зможе прогнозувати подальші користувацькі дії і попереджати можливі помилки на етапі конструювання схем бізнес-процесів.

Для переходу на програмне забезпечення від користувача не потрібно будь-яких додаткових навичок або знань, адже користувач бачитиме лише кінцевий результат роботи, а не керувати процесом, тобто прогнозування в програмному забезпеченні буде повністю автоматизовано.

5.4 Бізнес-продукт. Основні характеристики бізнес-продукту

Результатом виконання магістерської роботи є програмне забезпечення, яке дозволяє прогнозувати користувацькі дії при конструюванні схем бізнес-процесів виробництва. Це програмне забезпечення є доповненням результатів моєї бакалаврської роботи, а саме конструктора схем бізнес-процесів на платформі веб. З основних доповнень можна виділити: модуль, що прогнозує користувацькі дії і графічний модуль відображення результатів прогнозування.

В основу прогнозуючого модуля покладено дерево прецедентів, тобто структура даних, яка дозволить зберігати і зчитувати опис користувацьких дій, згруповані таким чином, щоб забезпечити облік хронології вчинення дій в залежності від ініціюючої дії.

На підставі вищезазначеного, можна заявити, що прогнозуючий модуль підтримує можливість самонавчання при обліку користувальницьких дій, тобто підлаштовується під роботу конкретного користувача.

Для повноцінної роботи наукового продукту необхідно розробити набір правил і вимог, на підставі результатів роботи якої можна з упевненістю стверджувати, що проміжна схема бізнес-процесів не містить помилок.

Таким чином, можна визначити, що підсумковим бізнес-продуктом є програмне забезпечення, яке дозволяє конструювати схеми бізнес-процесів і прогнозувати користувацькі дії на етапі конструювання.

Основними перевагами описуваного продукту перед аналогами є:

- можливість прогнозування користувацьких дій;
- надання допомоги користувачеві при конструюванні схем бізнес-процесів;
- виконання вищезазначених дій на етапі конструювання схем бізнес-процесів.

Підсумувавши вищезазначене можна вказати основні переваги, які користувач отримає при використанні описуваного бізнес-продукту:

- можливість конструювання схем бізнес-процесів в кроссплатформеному середовищі;
- отримання миттєвого аналізу схеми бізнес-процесів, що конструюється;
- зменшення кількості помилок, що допускаються, при конструюванні схеми;
- економія часових і матеріальних ресурсів.

5.5 Конкурентні переваги продукту

Визначимо і класифікуємо конкурентні переваги зазначеного продукту на дві категорії: природні (що мають основну перевагу) і штучні (маніпуляційні).

До категорії природних конкурентних переваг можна віднести наступні:

- наявність прогнозуючого модуля;
- незалежність від платформи виконання;
- вартість.

Основною конкурентною перевагою продукту є наявність модуля, що дозволяє прогнозувати користувацькі дії і аналізувати схему бізнес-процесів на наявність помилок, тим самим зводячи їх кількість до мінімуму.

Також важливим є можливість роботи з продуктом в браузері, тобто незалежно від середовища виконання, що дозволяє продукту бути гнучким у використанні, а користувачеві мати можливість конструювання схем бізнес-процесів з будь-якого пристрою, підключеного до мережі Інтернет.

Завершальною природною конкурентною перевагою є мінімальна вартість продукту, що дозволить користувачу налагодити процес конструювання схем бізнес-процесів без значних матеріальних витрат.

Перейдемо до категорії штучних конкурентних переваг, якими володіє вказаний продукт:

- можливість динамічного конструювання схем бізнес-процесів;
- можливість використання тимчасової підписки.

Під можливістю динамічного конструювання схем бізнес-процесів я маю на увазі необмеженість в наповненні конкретного рівня схеми, що дозволяє схемі бути гнучкою при конструюванні.

Можливість використання тимчасової підписки дозволить користувачеві спершу випробувати запропоновану функціональність продукту, а лише потім вирішити, чи варто йому купувати його або звернутися до конкурентних продуктів.

5.6 Клієнти. Сегменти ринку споживання

Для досягнення максимальної ефективності діяльності команди проекту рекомендується провести сегментацію ринку клієнтів, тобто виділити більш-менш однорідні групи користувачів, зацікавлених в пропонованому продукті.

Таким чином, сегментацію ринку клієнтів для конструктора схем бізнес-процесів можна провести за двома категоріями:

- обсяг запланованих робіт;
- можливість купити ліцензію на використання продукту.

Згідно з цими двома критеріями, потенційних клієнтів можна розділити на 2 групи:

- користувачі, які готові заплатити гроші за ліцензію на використання продукту, тим самим зняти обмеження на доступний обсяг робіт і рівень їх вкладеності;
- користувачі, які готові переглядати рекламу і обмежуватися обсягом робіт і рівнем їх вкладеності під час роботи з продуктом.

До першої категорії користувачів відносяться клієнти, професійна діяльність яких, переважно, пов'язана з проектуванням операційної діяльності – ця категорія включає в себе користувачів, які є основним джерелом доходу для компанії.

До другої, відповідно, користувачі, які знайомляться або навчаються проектування операційної діяльності.

5.7 Унікальна ціннісна пропозиція

Продукт, який я пропоную – це конструктор схем бізнес-процесів, доповнений можливістю прогнозування користувацьких дій під час конструювання.

Продукт корисний при конструюванні схем бізнес-процесів проектованої системи тим, що скорочує тимчасові і матеріальні витрати на процес конструювання.

5.8 Доходи і витрати

Оцінюючи витрати на проект, варто виділити дві основні категорії: стартові і щомісячні витрати. Стартові витрати – це гроші, які вкладаються один раз на початку проекту, а щомісячні, відповідно, вкладаються в розвиток проекту щомісяця.

Таким чином, в даному проекті до стартових відносяться витрати:

- на юридичні послуги;
- на ріелтерські послуги, мінімальні меблі, канцелярію;
- на комп'ютери та комплектуючі, ПЗ;
- на транспортні витрати.

Щомісячні витрати на проект поділяються на наступні групи:

- зарплати за посадами;
- оренда офісного приміщення;
- комунікації;

- податки;
- непередбачені витрати.

Загальні витрати на проект наведено в таблиці 4.

Таблиця 4. Загальні витрати на проект

Вид витрат	Вартість (\$)
Стартові	
Юридичні послуги	350
Рієлтерські послуги, мінімальні меблі, канцелярія	2100
Комп'ютери та комплектуючі, програмне забезпечення	3500
Транспортні витрати	150
Щомісячні	
Зарплати за посадами	2500
Оренда офісного приміщення	450
Комунікації	350
Податки	300
Непередбачені витрати	300

Таким чином, підсумувавши сказане, можна сказати, що сума, необхідна для старту проекту становить 6100\$, а щомісячні витрати складуть 3900\$.

Узагальнено кажучи, можна виділити дві стратегії отримання прибутку в веб-додатках: від користувачів і від рекламодавців. Для того щоб заробити якомога більшу кількість грошей на продукті, необхідно об'єднати обидві стратегії. У випадку з конструктором бізнес-процесів це можна

зробити створивши кілька типів ліцензій для використання програмного забезпечення.

Ліцензія, яка призначається для використання в корпоративних цілях (проектними менеджерами, інвесторами і т.д.), буде надавати повну функціональність системи і не буде демонструвати абсолютно ніякої реклами. Вартість ліцензії такого типу буде розподілена на тимчасові проміжки використання. Вартість одного часового проміжку використання продукту становитиме 20\$.

Ліцензія, яка призначається для використання не в корпоративних цілях, буде надавати обмежену функціональність проекту у вигляді обмежень на обсяги схем бізнес-процесів, наповненості і т.д. Також користувачеві буде демонструватися реклама. Поширюватися така ліцензія буде безкоштовно, але отримувати дохід на її використанні ми буде за рахунок переглядається реклами.

Необхідно, також, ввести тимчасову ліцензію (trial), яка не обмежена за функціональністю, але обмежена за часом використання. Призначатися цей тип ліцензії буде для ознайомлення з можливостями продукту.

5.9 Висновки

Опис бізнес-моделі продукту оформлено в форматі основних блоків, які надалі буде згруповано в коротку та ємку графічну схему бізнес-моделі.

Ціннісна пропозиція. Продукт, що пропонується – це конструктор схем бізнес-процесів виробництва з можливістю прогнозування користувацьких дій. Продукт призначається для комерційних організацій, що займаються проектуванням операційної діяльності. Продукт корисний тим, що дозволяє економити матеріальні та часові ресурси при конструюванні схем бізнес-процесів.

Відміни, що ведуть до переваг. Основною перевагою продукту є наявність прогнозуючого модуля, що дозволяє прогнозувати користувацькі дії під час конструювання схем бізнес-процесів. Використання зазначеного модуля дозволяє економити часові та матеріальні витрати на конструювання схем бізнес-процесів.

Сегменти споживачів. Споживачами даного продукту є люди, професійна діяльність котрих пов'язана з проектування операційної діяльності. Додатковою категорією споживачів є люди, що знайомляться та навчаються зазначеній сфері діяльності.

Канали продажів. Основний канал продажів, що існує у продукту – це власний продаж ліцензій на використання користувачеві. Також існує додатковий канал продажів – продаж місця для контекстної реклами. Основний дохід компанії забезпечує саме продаж ліцензій кінцевому користувачеві.

Структура доходів компанії. Вхідний потік коштів базується на періодичній оплаті за використання продукту. Іншими словами, користувач купує ліцензію і працює з продуктом оплачений час, без будь-яких обмежень на обсяги виконаних робіт.

Ключові партнери. Основні ресурси, необхідні для початку роботи - матеріальні інвестиції, розраховані на стартові витрати проекту і на піврічну діяльність (щомісячні витрати). Також важливим є партнерство з сервісами, які надають сервера для розміщення продукту.

Ключові ресурси. Для початку роботи і подальшого розвитку продукту необхідні фінансові ресурси, які дозволять найняти людські ресурси (фахівці в галузі проектування операційної діяльності, розробки та маркетингу), придбати комп'ютерне обладнання для розробки продукту і орендувати офісне приміщення.

Структура витрат. Основні витрати, які необхідні для розвитку проекту, є виплата заробітної плати співробітникам, підтримання комунікацій між співробітниками і серверної службою, а також витрати на оренду офісного простору.

Ключові вимірювачі. Ключовими факторами, які будуть свідчити про розвиток проекту, є рентабельність продажів і розвиток корисної функціональності, що надається продуктом.

Кінцева схема бізнес-моделі проекту наведена в таблиці А.2 в Додатку А.

ВИСНОВКИ

Виходячи з загальної ідеї прецедентного опису об'єктів задача прогнозування користувацьких дій при конструюванні схем бізнес-процесів зводиться до динамічного формування моделі поведінки конкретного користувача та подальшого аналізу побудованої моделі. Прецедентний опис ознак вважається потужною технологією, порівняно з іншими методами аналізу, які використовувалися раніше, чим дозволяє значно скоротити аналіз інформації в задачах високих розмірностей.

Беручи до уваги користувацькі вимоги, що були висунуті до способу та програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва, можна зазначити що спосіб прогнозування є унікальним та повністю дозволяє реалізувати всі поставлені до нього вимоги, а саме: використання власної активності користувача для побудови майбутніх прогнозів та висока точність прогнозів.

Для реалізації програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва буде використано клієнт-серверну архітектуру для додатку на платформі веб, що складається з двох основних частин, графічної та прогнозуючої, побудованих за модульною архітектурою та поєднаних між собою за допомогою програмного інтерфейсу взаємодії компонент через публічні методи класів.

За результатами тестування та аналізу кінцевого програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва можна заявити, що розроблений програмний застосунок не має помилок, які можуть призвести до виключних ситуацій при роботі користувача та повністю відповідає початковим вимогам, що

були висунуті в процесі анкетування потенційних користувачів програмного застосунку.

За результатами проведеної роботи було сформовано та побудовано бізнес-модель кінцевого продукту, що описує ключові моменти в організації діяльності, пов'язаної з поширенням розробленого програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва в комерційній середі організації проектної діяльності.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Распознавание. Математические методы. Программная система. Практические применения. – Журавлев Ю. И., Рязанов В. В., Сенько О. В. – М.: Фазис, - 2006;
2. Прикладный методы анализа данных и знаний. – Загоруйко Н.Г. – Новосибирск: ИМ СО РАН, - 1999;
3. Статистичне моделювання та прогнозування. – Єріна А.М. – К.:КНЕУ, - 2001;
4. Моделирование процессов выделения признаков с использованием латентно-семантического анализа. – Соловьев А.Н. – СПбГУ, - 2008;
5. Feature engineering for machine learning. – A.Zheng, A.Casazi, - O'Reilly, - 2018;
6. Прикладная статистика. – Орлов А.И. – Издательство «Экзамен», - 2004;
7. Введение в анализ данных. Учебник для бакалавриата и магистратуры – Миркин Б.Г. – М: Фазис, - 2016;
8. Управління виробничим підприємством [Електронний ресурс] // 1С. — 2016. — Режим доступу : http://1c.ua/v8/RegionalSolutions-UA_UPP.php.
9. Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE 2004 Version - Руководство к Своду Знаний по Программной Инженерии, в дальнейшем просто “SWEBOK” [SWEBOK, 2004];
10. Елиферов В.Г., Репин В.В. Бизнес-процессы. Регламентация и управление – издательство Инфра-М, Москва, 2005.
11. Аналіз вимог за Вігерсом [Електронний ресурс] // Business Analysis. — 2004. — Режим доступу : <http://iiba.ru/requirements-analysis/analysis-of-requirements-wiegers-2004/>.

12. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии// –М.: Изд-во МГТУ им. Н.Э. Баумана, 2005.
13. Прикладный методы анализа данных и знаний. – Загоруйко Н.Г. – Новосибирск: ИМ СО РАН, - 2005;
14. Методики збору вимог до програмного забезпечення [Електронний ресурс] // CASEClub . — 2003. — Режим доступу : <http://www.caseclub.ru/articles/trebmethod.html>.
15. Тестування програмного забезпечення [Електронний ресурс] // Вікіпедія. — 2015. — Режим доступу : <https://goo.gl/3UW46Q>.
16. Достоверный бизнес-план [Електронний ресурс] // Egorfine. — 2017. — Режим доступу: <https://egorfine.com/ru/articles/business-planning/>
17. Машинне навчання [Електронний ресурс] // Розпізнавання. — 2016. — Режим доступу: <https://bit.ly/2Sp5vb5>.

ДОДАТКИ

Додаток А

Порівняльний аналіз зацікавлених сторін

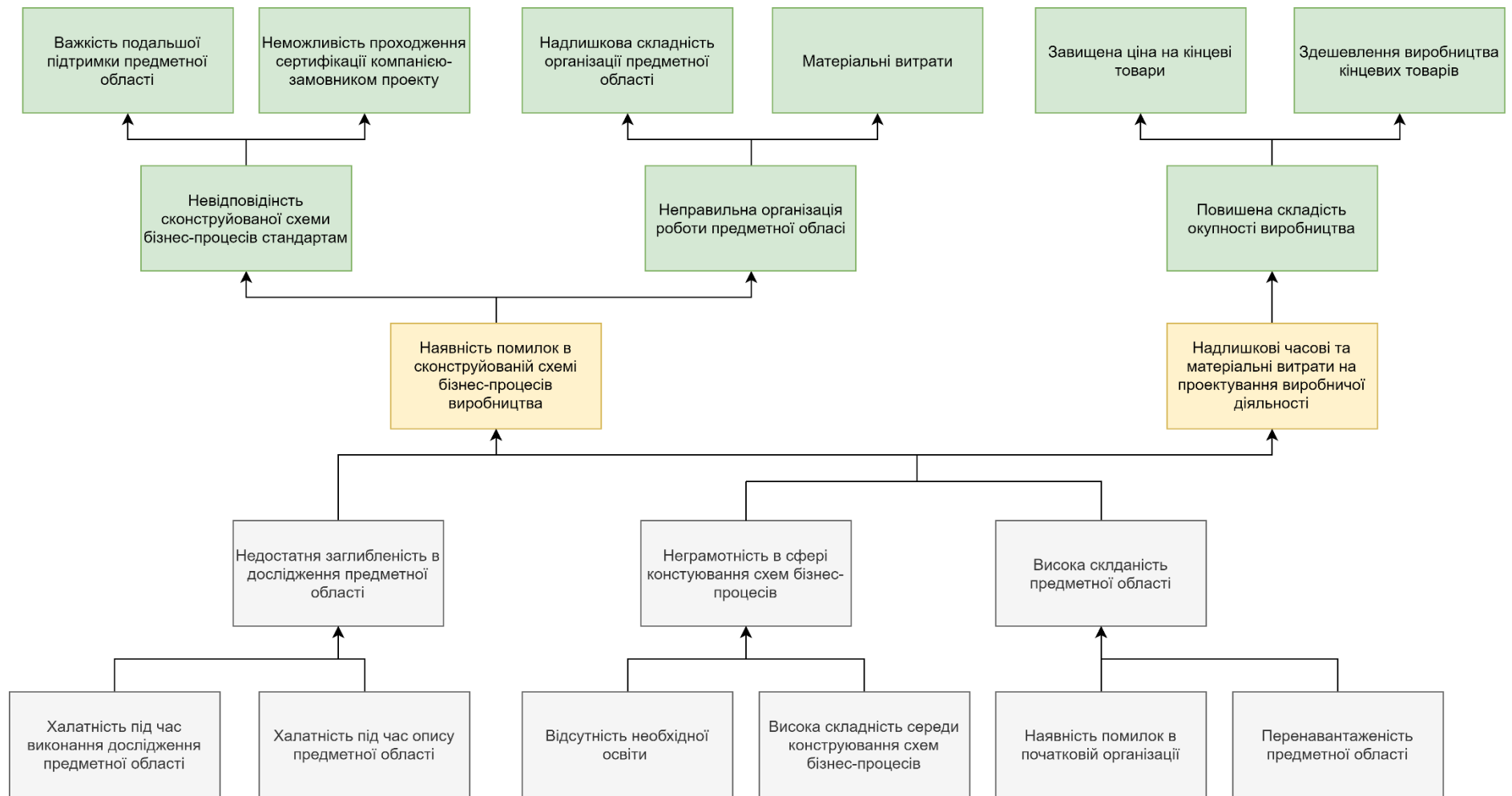


Рис. А.1 – Дерево проблем

Таблиця А.1. Порівняльний аналіз зацікавлених сторін

Зацікавлена сторона	Зацікавленість в проекті	Умови зацікавленості в довгостроковості проекту	Можлива участь
Проектний менеджер	Проект допомагає виправляти допущені помилки та прогнозує майбутні дії за для економії часу, необхідного на проектування виробничої діяльності	Проект допомагає знайти до 95% помилок в схемі бізнес-процесів та прогнозує користувацькі дії з точністю не менше 90%	Цільова група
Інвестор	Проект допомагає економити час і гроші, необхідні на проектування виробничої діяльності	Проект допомагає економити час і гроші, необхідні на проектування виробничої діяльності	Надання інвестицій на розробку проекту
Виконавець спроектованого проекту	Упевнений, що помилок в спроектованій системі не допущено, а значить сильніше зацікавлений в його реалізації	Проект допомагає економити час і гроші, необхідні на реалізації спроектованої системи	-
Конкуренти	Зацікавлений в тому, щоб проект закритися, тому що проект має більшу кількість корисної функціональності, ніж його власний	-	Створення перешкод на шляху реалізації проекту

Таблиця А.2. Схема бізнес-моделі

Програмне забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва				
Ключові партнери - інвестор; - сервіси, що надають сервера для розміщення продукту.	Ключові вимірювачі - рентабельність продажів; - розвиток корисної функціональності продукту.	Ціннісна пропозиція - конструктор схем бізнес-процесів; - виявлення користувацьких помилок; - прогнозування користувацьких дій.	Відміни, що ведуть до переваг - прогнозуючий модуль; - зменшення кількості помилок на 95% та прогнозування з точністю 90%.	Сегменти споживачів - проектні менеджери; - люди, що знайомляться з проектування операційної діяльності.
	Ключові ресурси - фінансові ресурси; - людські ресурси; - комп'ютерне обладнання.		Канали продажу - продаж ліцензій кінцевому користувачеві; - продаж місця для реклами.	
Структура витрат - виплата заробітної плати співробітникам; - підтримка комунікації між співробітниками та серверами; - оренда офісного приміщення.			Структура доходів - періодична плата за використання продукту; - плата за розміщення реклами.	

Додаток Б

Код розробленого програмного забезпечення

Файл package.json

```
{  
  "name": "mwp",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "react": "^16.6.3",  
    "react-dom": "^16.6.3"  
  },  
  "devDependencies": {  
    "babel-core": "^6.26.3",  
    "babel-preset-env": "^1.7.0",  
    "babel-preset-react": "^6.24.1"  
  }  
}
```

Файл index.html

```
<html>

<head>

  <title>Application</title>

</head>

<body>

  <div id="root"></div>

  <script src="./js/main.js"></script>

</body>

</html>
```

Файл main.js

```
import React from "react";
import {render} from "react-dom";
import App from "./App.jsx";

import "../css/index.css";

render(
  <App/>,
  document.querySelector("#root")
)
```

Файл index.css

```
html, body {  
    height: 100%;  
    margin: 0;  
  
    font-family: Arial;  
}
```

```
h1 {  
    color: #414141;  
}
```

```
.wrapper {  
    display: flex;  
    flex-direction: row;  
    height: 100%;  
  
    padding: 50px 0;  
    box-sizing: border-box;  
}
```

```
.actions {  
    width: 40%;  
    padding: 0 20px;
```

```
border-right: 1px solid #414141;  
}
```

```
.available {  
    padding-bottom: 20px;  
    border-bottom: 1px solid #414141;  
}
```

```
.history {  
    width: 60%;  
    padding: 0 20px;  
}
```

```
.history .item:not(:last-of-type):after {  
    content: "";  
    position: absolute;  
  
    width: 10px;  
    height: 10px;  
    right: -17px;  
  
    background: url("./arrow.svg") no-repeat center;  
}
```

```
.item {  
    position: relative;  
    width: 50px;  
    height: 50px;  
  
    display: inline-flex;  
    align-items: center;  
    justify-content: center;  
  
    box-sizing: border-box;  
    border: 2px dotted #414141;  
  
    cursor: pointer;  
}
```

```
.item--red {  
    background-color: rgba(255, 0, 0, 0.5);  
}
```

```
.item--green {  
    background-color: rgba(0, 255, 0, 0.5);  
}
```

```
.item--blue {  
    background-color: rgba(0, 0, 255, 0.5);  
}
```

```
.item--yellow {  
    background-color: rgba(255, 255, 0, 0.5);  
}
```

```
.item:not(:last-of-type) {  
    margin-right: 20px;  
    margin-bottom: 20px;  
}
```

```
.history .item:last-of-type {  
    border: 2px solid #414141;  
}
```

```
.unhovered {  
    opacity: 0.2;  
}
```

Файл App.jsx

```
import React, { Component } from "react";

const Item = ({ color, id, className = "", onClick, onMouseEnter, onMouseLeave })
=> {

  return (

    <div className={['item item--${color}', className].join(" ")}

      onClick={onClick}

      onMouseEnter={onMouseEnter}

      onMouseLeave={onMouseLeave}

    >{id}</div>

  );

};

export default class App extends Component {

  constructor(props) {

    super(props);

    this.actions = [

      { id: 0, color: "red" },

      { id: 1, color: "yellow" },

      { id: 2, color: "green" },

      { id: 3, color: "blue" },

    ];

  }

}
```

```

    this.state = {
        history: [],
        hovered: null
    };

    this.makeAction = this.makeAction.bind(this);
}

makeAction(action) {
    this.setState(ps => {
        return Object.assign({}, ps, {
            history: ps.history.concat(action)
        })
    })
}

render() {
    const { history, hovered } = this.state,
        lastAction = history[history.length - 1],
        lastActionId = lastAction ? lastAction.id : null;

    return (
        <div className="wrapper">
            <div className="actions">

```



```

<div className="available">
  <h1>Доступні дії</h1>
  <div className="items">
    {
      this.actions.map(action => (
        <Item {...action}
          onClick={() =>
this.makeAction(action)}
        />
      ))
    }
  </div>
</div>

<div className="prognosed">
  <h1>Спрогнозовані дії</h1>
  <div className="items">
    {
      history.reduce((res, item, index,
array) => {
        if (item.id === lastActionId
&& index + 1 < array.length) {
          let newElem =
array[index + 1],
          existedIndex =
res.findIndex(_ => _.id === newElem.id);
          if (existedIndex >= 0)
{

```

```

return
res.slice(0, existedIndex)

.concat(

Object.assign(

{ },

res[existedIndex],

{priority: res[existedIndex].priority + 1}

)

)

.concat(res.slice(existedIndex + 1, res.length));

} else {

return

res.concat(Object.assign({ }, newElem, {priority: 1}));

}

} else {

return res;

}

}, [])

.sort((a, b) => b.priority -

a.priority)

.map(item => <Item

{...item}

```

```

onClick={_ => {
    this.setState(ps => Object.assign({ps}, {
        hovered: null
    })), () => this.makeAction(item))
}}
onMouseEnter={_ => this.setState(ps =>
Object.assign({ps}, {
    hovered: item.id
    })))}

onMouseLeave={_ => this.setState(ps => Object.assign({ps}, {
    hovered: null
    })))}

/>)
}
</div>
</div>
</div>
<div className="history">
    <h1>Історія викликів</h1>
    <div className="items">
        {
            history.map((item, index, array) => {
                let prevItem = array[index - 1];

```

```

let nextItem = array[index + 1];

return (
  <Item {...item}
    className={
      hovered ===
null ? "" : (
        (
          (
            (item.id === lastActionId) &&
            ((nextItem && nextItem.id) === hovered)
          ) ||
          (
            (item.id === hovered) &&
            ((prevItem && prevItem.id) === lastActionId)
          ) ||
          (
            index === array.length - 1
          )
        ) ? "" :
"unhovered"
      )
    }
  )
}

```

```
}  
}  
};  
  
</div>  
  
</div>  
  
</div>  
}  
});  
</div>  
</div>  
</div>
```

Список

опублікованих наукових праць Шевчука Євгенія Олеговича
за темою магістерської дисертації:

«Спосіб та програмне забезпечення прогнозування користувацьких дій при
конструюванні схем бізнес-процесів виробництва»

1. Шевчук Є.О. Програмно-аналітичний модуль прогнозування користувацьких дій при конструюванні схем бізнес-процесів/ Є.О. Шевчук, Л.А.// XI наукова конференція магістрантів та аспірантів «Прикладна математика та комп'ютинг (ПМК-2018-2) : збірник тез доповідей. – К. : ФПМ КПІ ім. Ігоря Сікорського, 2018. – 4 с.

Студент _____
(підпис)

Шевчук Є.О. _____
(прізвище, ім'я, по батькові)

Науковий керівник _____
(підпис)

ст. викладач, к.т.н. Люшенко Л.А. _____
(посада, вчене звання, науковий ступінь, прізвище, ініціали)

РЕЦЕНЗІЯ
на магістерську дисертацію
на здобуття ступеня магістра-професіонала

виконаної на тему: СПОСІБ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ
СХЕМ БІЗНЕС-ПРОЦЕСІВ ВИРОБНИЦТВА
студентом (-кою) Шевчуком Євгенієм Олеговичем
(прізвище, ім'я, по батькові)

Магістерська дисертація, представлена для рецензування, за змістом повністю відповідає затвердженій темі та завданню. Розробка способу та програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів є актуальною задачею та має практичну цінність, оскільки дозволяє економити часові та матеріальні ресурси, необхідні на реалізацію проекту розробки програмного забезпечення.

В магістерській дисертації проаналізовано актуальні задачі, які виникають при конструюванні бізнес-процесів для реалізації програмного забезпечення. Зібрано користувацькі вимоги до програмного забезпечення, детально описано процес розробки способу прогнозування користувацьких дій та програмного забезпечення, що реалізує зазначений спосіб. Якісно проаналізовано результати роботи та запропоновано бізнес-модель для реалізації розробленого програмного забезпечення.

Вважаю, що магістерська дисертація студента Шевчука Є.О. відповідає вимогам до випускних кваліфікаційних робіт освітньо-кваліфікаційного рівня «Магістр-професіонал» та заслуговує відмінної оцінки, а її авторів, студенту Шевчуку Є.О., може бути присвоєна кваліфікація «Професіонал у галузі програмування».

Рецензент

доцент, к.ф.-м.н., доцент
(посада, науковий ступінь, вчене звання)

(підпис)

І.А. Шубенкова
(ініціали, прізвище)

Печатка установи, організації рецензента *(тільки для зовнішнього рецензента)*

ВІДГУК
керівника магістерської дисертації
на здобуття ступеня магістра-професіонала

виконаної на тему: СПОСІБ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ
СХЕМ БІЗНЕС-ПРОЦЕСІВ ВИРОБНИЦТВА
студентом (-кою) Шевчуком Євгенієм Олеговичем
(прізвище, ім'я, по батькові)

Магістерська дисертація студента Шевчука Є.О., присвячена вирішенню проблеми пришвидшення конструювання релевантних бізнес моделей програмного забезпечення за рахунок автоматизації процесу прогнозування користувацьких дій при конструюванні схем бізнес-процесів. Магістерська дисертація виконана за наукової тематикою кафедри програмного забезпечення комп'ютерних систем. Випускна кваліфікаційна робота Шевчука Є.О. повністю відповідає виданому завданню.

Під час виконання роботи студент Шевчук Є.О. продемонстрував ґрунтовні знання, старанність, здатність та вміння застосовувати набуті знання для вирішення складних завдань.

Особистим здобутком Шевчука Є.О. у рамках дипломного проекту є розробка способу та програмного забезпечення прогнозування користувацьких дій, що ґрунтується на попередній активності користувача при роботі із програмним забезпеченням.

На мою думку, якість підготовки Шевчука Є.О. відповідає вимогам освітньо-кваліфікаційної характеристики спеціальності 6.050103 «Програмна інженерія». Вважаю, що випускна кваліфікаційна робота відповідає вимогам до магістерських дисертацій, а її автору, Шевчуку Є.О., може бути присвоєна кваліфікація «Професіонал в галузі програмування».

Унікальність магістерської дисертації рівняється _____.

Керівник магістерської дисертації

старший викладач, к.т.н., доцент _____ Л.А. Люшенко